

3192

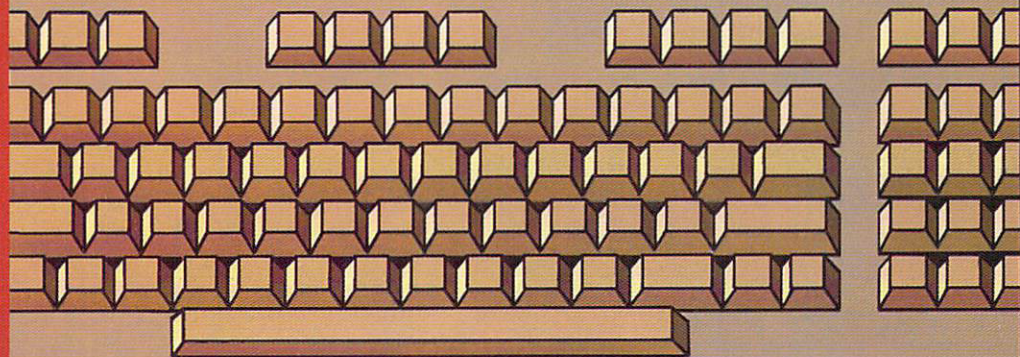
TIPS & TRICKS



FOR COMMODORETM COMPUTERS

"a must-buy for any Commodore computer user"
"an indispensable reference"

Jim Gracely, Managing Editor
Commodore Magazine



Louis F. Sander

LOU SANDER'S TIPS & TRICKS FOR COMMODORE COMPUTERS

LOUIS F. SANDER

"a must-buy for any Commodore computer user"
"an indispensable reference"

Jim Gracely, Managing Editor
Commodore Magazine

 WINDCREST

Commodore 64®, **CBM®**, **PET®** and **VIC 20®** Commodore Electronics, Ltd.
Commodore 128™ Commodore Electronics, Ltd.
Amiga® Commodore-Amiga.
IBM® International Business Machines Corp.

Published by **Windcrest Books**
FIRST EDITION / SECOND PRINTING

© 1989 by **Windcrest Books**. Reproduction or publication of the content in any manner, without express permission of the publisher, is prohibited. The publisher takes no responsibility for the use of any of the materials or methods described in this book, or for the products thereof.

Library of Congress Cataloging-in-Publication Data

Sander, Louis F.
Lou Sander's tips and tricks for Commodore computers.

Includes index.

1. Commodore computers—Programming.

I. Title II. Title: Tips and tricks for Commodore computers.

QA76.8.C64S257 1989 005.265 88-35635

ISBN 0-8306-9192-8

ISBN 0-8306-3192-5 (pbk.)

TAB BOOKS Inc. offers software for sale. For information and a catalog, please contact TAB Software Department, Blue Ridge Summit, PA 17294-0850.

Questions regarding the content of this book should be addressed to:
Windcrest Books
Division of TAB BOOKS Inc.
Blue Ridge Summit, PA 17294-0850

Ron Powers: Director of Acquisitions
David M. Harter: Technical Editor
Katherine Brown: Production

Contents

<i>Acknowledgments</i>	xx
<i>Introduction</i>	xxii
1 Learning	1
Resources for Learning	2
Reader's Guide to Computer Literature	3
Read It and (Don't) Weep	4
Two Books Are Better than One	4
Go to the Meetings!	4
Finding Fellow Users	5
A Place to Start Your User Group	5
Finding User Group Meeting Rooms	6
The Ultimate Tip	6
Related Tips in Other Chapters	6
2 Programs from Books and Magazines	7
Typing In Programs	8
Program Typing Tip	9
Magazine Placekeeper	9
Read It to the End	10
Label the Programs You Type	10
Corrections to Published Programs	10
Keystroke Descriptions in Listings	11
Commodore Magazine's Magazine Entry Programs	13
Magazine Entry Program Hints	14
Magazine Entry Program Bypass	14
Belated Use of the 128 Magazine Entry Programs	15
Magazine Entry Program Improvements	15
Magazine Entry Program-128 Speedup	15

Modifying the Magazine Entry Programs 16
Modified 64 Magazine Entry Program 16
Modified 128 Magazine Entry Program 17
MEP Autoloader 18

3 Saving Money 20

Finding Computer Bargains 21
Free Paper, Free Disks, Free Equipment! 21
Enhancing Your Warranty 22
Tax Tips 22
Saving Money on Printed Matter 23
Classified Ads Save Money 23

4 The Computer Room 25

Glare-free Lighting 26
Easy Printer Stand 27
Monitor Shelf 27
Lightning Detector 28
Connector Labels 28
Handy Notes 29
“SAVE YOUR MARRIAGE”,8 29
Nice Marker 29
Printer Ruler 29
Household Items in the Computer Room 30
Inexpensive Cleaning Fluid 30
Printer Scrap to Your Notebook 31
Printer Scrap to the Microwave 31

5 Keys and Keyboarding 37

Slow Listing 33
The “Don’t Do It” Key 33
Easy RUNning 33
More Easy RUNning 33
Typing Large Numbers 34
Abbreviating Zero 34
BASIC Abbreviations 34
Watch Those 128 Abbreviations! 35
Typing Similar Lines 36
Keyboard Curiosity 36
Keyboard Cleaning 36
Readable Multicolor Characters 37

All About the Stop and Restore Keys	37
Disabling the Stop and Restore Keys	39
Disabling the 64 Restore Key	39
64 Restore/Reset Disable	40
64 Delete Right	40
64 F7 Apostrophe Key	41
128 KEY Hint	41
128 F-key Keeper	41
128 Auto LOAD/RUN	42
128 F-Key Conversions	42
Ultimate F-key Disabler	43
128 Caps Lock Reader	43
Quick Escape from 128 Graphics	44
Another 128 ESCape Sequence	44

6 Video Displays **45**

Tips for Better Video	46
Shield Your TV Cables	47
Modify Your TV Switchbox	47
Monitor Glare Reduction	48
Sight Improver	48
Screen Marking Hint	48
Screen Photography	49
RGB Monitor Extension	49
1902 Monitor Tips	49
128 Video Monitor Switch	50

7 Printers and Printing **51**

Printer DIP Switches	52
Power for Your Interface	52
Using Multiple Printers	53
Two Printers with One Computer	54
Printer Pause Control	54
Printer Hex Mode	54
Extra Print Modes	55
Printer Paper Alignment	55
Feeding Bottom Feeders	56
Easier Printer Control	56
MPS803 Tractor Modification	57
MPS1000 in NLQ	58
Blue Chip Printers	58
Okimate Head Cleaning	58

8 Printer Supplies

60

- Mark Your Fanfold Paper 61
- Home-made Fanfold 61
- Paper Saver 61
- Inexpensive Heat-sensitive Paper 62
- Better Paper for Okimate Color Printing 62
- Labels and Gummed-up Printers 62
- Cheap Printer Labels 63
- Colorful Printouts 63
- Home-made Envelopes 63
- Greeting Card Envelopes 64
- Keep Two Ribbons 64
- MPS801 Ribbons 64
- MPS803 Ribbons 65
- Daisy Wheel Ribbons Explained 65
- Emergency Ribbon Replacement 65
- New Life for Old Ribbons 66
- Rejuvenating Cloth Ribbons 66
- Ribbon Re-inking 66
- 1520 Plotter Supplies 66
- DPS1101 Printwheels 67
- More on DPS1101 Printwheels 68

9 Of Mice and Joysticks

69

- 1350 Mouse as Joystick 70
- 1351 Mouse in Joystick Mode 70
- Label Your Joysticks 70
- Use the Proper Joystick 70
- Troubleshooting Joysticks 71
- Joysticks and Keyboard Errors 71
- Using Non-Commodore Joysticks 72
- Joysticks for Southpaws 72

10 Datassette Recording

73

- Datassette Tips 74

11 Floppy Disks

76

- Disk Buyer's Guide 77
- How to Make Flippy Disks 78
- Flippy Disk Identifier 79
- Assigning Disk ID's 79

Disk Library Tips 79
Disk Tabs and Labels 80
Better Disk Labels 81
Home-made Disk Envelopes 81
Colorful Disk Envelopes 82
Disk Mailers 82
Thick Diskettes 82
Creative Disk Storage 83
Disk Storage 83
Temporary Disk Storage 83
64C and 128 Temporary Disk Storage 84

12 Disk Operations

85

Temporary Program Names 86
Program Naming System 86
Data and Time in Filenames 87
TI\$ in Filenames 87
Another TI\$ SAVER 88
First Program in the Directory 88
Loading from the Directory 89
Easy-load Filenames 89
128 Easy LOAD and RUN 90
Disk Directories by Filetype 90
Making Directory Printouts 91
Double-spaced Directory Printouts 91
Improved Directory Printouts 91
Reading Protected Directories 92
Backup Disks and Heartbreaks 92
Easy Disk Drive Reset 93
Head Stick Fix 93
Disk Head Unsticker 94
Transporting Disk Drives 94
Avoiding Loading Slowdown 94
64 DOS Wedge Slows Programs 95
128 DOS Shell Copies 95
128 DOS Shell Speedup 95
128 and Locked Files 96
1571/64 Speed Report 96
1571 Storage Limitation 96
Double-sided 1571 in 64 Mode 97
1571 Quirks 97
1571 Motor Shutoff 98

13 Telecommunications 100

- Modem Connections 101
- Modem Speed Dialing 101
- Telephone Extensions 101
- Marking Telephone Cables 101
- Telecommunications Tips 102
- 1670 Autoanswer Disabler 102
- 1670 Silencer 103
- Disabling Modem Autoanswer 103
- Modem Hang Up 104
- Telecommunications Privacy 104
- Online Recording 104
- Minimizing Telecommunications Charges 105
- Online Assistance 105
- Online Reading Practice 106
- Meet Me on Quantum Link 106
- Quantum Link Timesaver 106
- Quantum Link Quick Messages 107
- Quantum Link File Append 107
- Quantum Link Multitasking 107
- Quantum Link Rosters 108
- Quantum Link Repeating Keys 108
- Quantum Link Screen Names 108
- 128 Quantum Link Password 108
- GREAT Programs 109
- Tips on Running a BBS 109

14 Wiring and Connectors 111

- Label That Cable 112
- DIN Connectors 112
- Joystick Port Connectors 113
- 128 Inaccessible Joystick Ports 113
- User Port Connectors 113
- Connector Polarization 114
- Expansion Port Connectors 114

15 Troubleshooting 116

- Hang-up Recovery 117
- 128 Apparent Failure 117
- Reviving Dead Computers 117

Power Supply Problems 118
Dead 128 119
Schematic Diagrams 119

16 A Hardware Assortment 121

All About the Reset Button 122
128 Reset Button Extender 124
Resetting the 128, Plus/4 and C16 125
Disconnection Discombobulation 124
Beating the Serial Bus Blues 125
Cartridge Tips 126
Use a Cartridge Expander 126
Cartridge Lifesaver 127
RAM Expander RAMDOS 127
Become a Hardware Hacker 127
Joystick Pieces 128
Burning In 128

17 Other Computers 130

Generic Tips and Tricks 131
Amiga Border Remover 131
Amiga Default Colors 131
IBM/Commodore Compatibility 131
Notes from PC Land 132
Plus/4 Resources 133
Plus/4 Documentation 134
Plus/4 Items in this Book 135
Plus/4 Graffiti 135
Plus/4 Machine Language Storage 135
Plus/4 and C16 Line Width 135
SX 64 Keyboard Repairs 135
SX 64 and Printer Interfaces 136

18 Learning to Program 137

The Way We Were 138
Why You Should Learn BASIC Programming 139
How to Learn BASIC 140
Learning BASIC 141
Comparative Anatomy 142
Learning Machine Language 142

19 The BASIC Programming Language

144

AUTO Fix 145
Which BANK? 145
Using BEGIN and ELSE 145
CHAR Trickery 146
CONT for Endless Loops 146
Shortening DATA Statements 146
DLOAD and Other Disk Commands 147
DRAW Shortcut 147
DSAVE with Replace 148
Happy ENDings 148
FOR...NEXT Hints 149
Flashing Cursor for GETKEY 150
Shorter IF Statements 150
INPUT without a Question Mark 151
INPUTting Illegal Characters 151
INPUT with Default 152
Unwanted Characters in INPUT 153
INT Hints 154
INTEResting Bug 154
Double-spaced LIST 154
Indented LIST 155
LIST Disablers 155
128 LIST Bell 156
PLAY Statement 156
What Are Those POKEs, Anyway? 157
PRINT and the Semicolon 157
Structured REMs 158
RENUMBER Trick 158
Disabling SAVE 159
Disabling 64 SAVE 159
Setting SPRDEF Background Colors 160

20 Machine Language

161

Printouts from Machine Language Monitors 162
Machine Language Monitors as Assemblers 162
128 Machine Language Monitor and Resets 163
128 Machine Language Monitor for 64 Mode 163
Supermon Revealed 163
Machine Language String Printer 164

Machine Language Screen Printing 165
Memory Maps 165
Digging into Machine Language 168

21 Other Languages 192

A Feast of Languages 193
COMAL Explained 194
Compiled COMAL 195
COMAL Turtle Graphics Tip 195
COMAL File Copier 195
LOGO and Print Shop 195
Super C Memory Access 196
Super C Bug 196
Super Pascal File and Variable Names 196
Super Pascal File Safety 196

22 The CP/M Operating System 197

What Is CP/M? 198
Locating CP/M Software 199
Kaypro CP/M Programs on the 128 199
128 CP/M Reboot 199
Changing CP/M Screen Width 199
CP/M Disk Swapping 200
128 CP/M and Printers 200

23 Software Potpourri 202

Bytes 203
Naming Variables 203
Temporary Line Deletion 204
LIST Stopper 204
Automatic Line Listing 204
Speed Testing 206
Faster Execution 206
Sensing Special Keys 207
Detecting Disabled Stop Key 207
Machine Type Detector 208
Editing SEQ Files 208
Efficient Numeric Files 209
Correcting Input 209
Character Format Conversions 209

- 64 Program Protection 211
- Defeating Program Protection 211
- Deleting 64 Program Lines 212
- 64 Line Number Suppression 212
- 64 POP Routine 212
- 128 Reset Changer 213
- 128 Line Counter 213

24 Programming the Video Display

214

- Centering Text 215
- Shaky Computing 215
- Screen Saver 215
- Color on a Monochrome Monitor 216
- Fancy Printing 216
- Changing Screen Color 218
- Changing Character Color 218
- Computer Shake 219
- Screen Scrambler 219
- Screen Framer 220
- 64 Screen Bars 221
- 64 Color Flipper 221
- 64 F-Key Colors 222
- 64 Color SYS 223
- 64 Easy Windows 223
- 64 PRINT@ 224
- 64 Screen Effect 225
- 64 Border Fireworks 225
- 128 Auto-Centering 226
- 128 Screen POKEs 227
- 128 80-column Line Space Changer 229
- 128 Simultaneous Text and Graphics 230
- 80-column Flashing and Underlining 230
- Pausing 128 Graphics 230
- 128 Symmetrical Scaling 230
- 128 Screen Mode Swapper 230
- 128 Sprite Rotator 231
- 128 Screen Explosions 232
- 128 Border Killer 233
- 128 End Writer 233

25 Programming Music and Sound

235

Buzzer Sound 236
Better Buzzer 236
Bell Sound 236
Bell Tone 237
Helicopter Sound 237
Beep Routine 237
Explosion Maker 238
European Siren 238
Charge! 239
Fanfare 239
Bugle Calls 240
Assembly 240
Taps 240
128 Assembly 240
128 Taps 240
Lullaby 240
128 Happy Birthday 242
128 Star Wars Theme 242
64 Keybeeper 243
128 Easy Player 244
ADSR Table Maker 244
128 TEMPO Table Maker 245
Musical Staff Display 246
Sheet Music 246

26 Mathematics and Programming

248

Numbers, Numerals and Bases 249
Telling Numerals Apart 250
Binary for Beginners 251
Binary and Hexadecimal 252
Decimal to Binary Converter 253
Binary to Decimal Converter 253
Decimal to Hex Converter 254
Hex to Decimal Converter 254
Universal Base Converter 255
128 Arabic to Roman Conversion 255
64 Binary Key Decoder 256
All About Rounding 257
Roots 258
Factorials 258

Permutations and Combinations 259
Logarithms 260
Working with Large Numbers 260

27 Debugging 262

Easier Bug-spotting 263
Let the Computer Read It 263
Finding Syntax Errors 263
Deceptive Lines 264
Unseen Characters in a String 264
Debugger's Nightmares 265
Debugger's Helper 266
Debugging Using STOP and CONT 266
Dump Those Variables 267
128 Variables 268
129 Editing Precaution 268
Debugging DATA Statements 268
TAB Errors 270

28 Translating 271

64 to 128 Translations 272
128 Programs and the 64 Super Expander 272
128 Sprites on the 64 272
64 Sprites on the 128 273
64 to PET Translation 274

29 Amusements and Delights 276

Mind Reader 277
Burrow 277
Potholes 277
Reach! 278
Flower Power 279
Mighty Mite 279
Lampman 281
Jaws LXIV 281
Jumbles 282
Piece de Resistance 283
Castle Siege 284
Whiz Bang 285
Psychological Insights 285

Fuzzy Caterpillar 286
Defensive Driving 287
Chopper Pilot 288
64 Mirror 288
64 Paratrooper 290
64 Sea Battle 291
64 Timetrial 292
64 Break-Fast 293
128 Screenshot 293
128 Hidden Message 295
128 Bird Art 295
128 Evasion 296
128 Commodore Logo 296
128 Error Messages 297
Grand Finale 297

30 Useful Application Programs

299

Lottery Assistant 300
Game Map Maker 300
Dice Machine 301
Sprite Graph Maker 302
64 Easy Word Processor 302
128 Easy Word Processor 304
Label Printer 305
Graph Paper Maker 305
Wind Chill Calculator 307
Fahrenheit to Celsius 307
Batting Averages 308
Alphabet Trainer 308
Division Tutor 310

31 Disk Utility Programs

311

File Renamer 312
Directory Viewer 312
Blocks Free Reader 313
Write-Protect Detector 313
Start Address Finder 313
End Address Finder 314
Memory Saver 314
64 ML Saver 315
64 Memory Saver 317

SEQ File Reader 317
128 SEQ File Reader 318
Disk Signer 318
64 Directory Basher 318
File Goofer 320
64 Disk Defaulter 320
64 Disk Error Reader 321
64 DOS Wedge Disabler 321
64 DOS Wedge Defaulter 322

32 Other Utility Programs 344

Video Test Pattern Generator 325
Sprite Finder 326
64 Sixpack Plus 326
64 Datamaker 329
64 Timer 329
64 Variable Dump 331
64 Pauser 333
64 Remvert 333
64 Quick Erase 334
64 Search 335
64 BASIC Linker 336
64 Machine Language Loader 338
64 Appender 339
128 Appender 340
128 Mode Guard 341
128 Hi-Res Helper 341
128 Restore Colors 342

33 Commercial Software: General Tips 349

Save Your Receipts 345
Software Documentation 345
Commercial Software and Your Printer 345
POKEs and Commercial Software 345
Commercial Software and the Reset Button 346
128 Keys in 64 Mode 346
Using an Extra Hand 346
Bypassing Dongles and Protection Keys 347
Using Key-Protected Software on the 128 347
Track and Sector Editors Explained 347

Acrojet	350
Aztec	350
The Bards Tale II Tip 1	350
The Bards Tale II Tip 2	351
The Bards Tale II Tip 3	351
The Bards Tale II Tip 4	351
The Bards Tale II Tip 5	352
The Bards Tale II Tip 6	352
Beach Head	352
Booty	352
Bop n' Wrestle Tip 1	353
Bop n' Wrestle Tip 2	353
Bruce Lee	353
Burnin' Rubber	353
Championship Lode Runner	354
Elite	354
F-15 Strike Eagle	354
Goonies	355
Gridrunner II	355
H.E.R.O.	355
ICUPS	356
Leader Board	356
Miner 2049'er	356
One-on-One	357
Pitfall	357
Popeye	357
Quest for Tires	357
Raid over Moscow	358
Rambo	358
Seven Cities of Gold	358
Space Harrier	358
Spy vs. Spy	359
Summer Games	359
Summer Games I Tip 1	359
Summer Games I Tip 2	360
Super Huey Tip 1	360
Super Huey Tip 2	360
Ultima III	360
Warhawk	361

Winter Games Tip 1 361
Winter Games Tip 2 361
Zaxxon 361

35 Other Commercial Software

363

Blitz! 364
Capture 364
Doodle Tip 1 364
Doodle Tip 2 364
64 Doodle Picture Loader 365
Easy Script Tip 1 365
Easy Script Tip 2 366
Easy Script Tip 3 366
Easy Script Tip 4 366
Fast Hackem 367
Fastload Cartridge Tip 1 367
Fastload Cartridge Tip 2 367
Fastload Cartridge 368
Fleet System 2 Tip 1 368
Fleet System 2 Tip 2 368
Fontmaster 368
General Ledger 369
GEOPAINT 369
GEOPUBLISH 370
GEOS Tip 1 370
GEOS Tip 2 370
GEOS Tip 3 370
GEOS Tip 4 371
GEOS Tip 5 371
GEOS Tip 6 371
GEOS Tip 7 372
GEOS Tip 8 372
GEOS Tip 9 372
GEOS 128 373
GEOWRITE 373
Hesmon 373
Koala Painter 374
Magic Desk 374
MicroIllustrator 374
Multiplan 373
The Newsroom 375

PaperClip Tip 1 375
PaperClip Tip 2 375
S'More Cartridge 376

36 Word Processing

377

Word Processing Placeholder 378
Convenient Word Processor Formatting 378
Unslashing Your Zeroes 378
Video Proofreading 378
Improving Justification 379
Automatic Paragraph Numbering 379
Printing Labels 380
Organizing Word Processor Files 380

Index

382

Acknowledgments

The most obvious contributors to this book are the creative and ingenious people whose names appear on its pages. But others have done their part in less conspicuous ways:

First, some special people whose inspiration allowed me to become a compiler of computer tips. Marcus and Horton's "Basic Radio" was the first technical book I ever owned. I understood only a quarter of it, but it planted the seeds for this volume. Ann H. Hartmann and H.P. Smith, teachers at West View High School, who taught me to think and to write. Margaret Ittel, student editor of the West Wind, who was the first to publish my writings. Thomas Dwyer and Margot Critchfield, whose book taught me to program in BASIC. Tom Fetherston, who showed me my first Commodore tip. Victor Pitre, without whose friendship and expert knowledge I would have remained a frustrated beginner. Gene Beals, whose PET User Notes were the inspiration for my first column. Guy Wright and Dennis Brisson, who recognized its potential.

After inspiration comes work—lots and lots and lots of it. Major patrons of that work have been Jim Gracely, Sue West, Julie Bauer, Diane Lebold, Carole Minton and Bob Kenney, all current or former executives at *Commodore Magazine*. Without their constant support, we wouldn't have "Tips & Tricks."

Special thanks are due to Barbara A. Sander, a reluctant and underpaid bookkeeper, who grits her teeth each month and writes dozens of checks to contributors.

Vitally important, of course, are the thousands of Commodore enthusiasts whose ideas and programs flow steadily into Box 101011. They provide not only raw material, but also constant encouragement. With truly amazing consistency, they express the latter as "keep up the good work."

Over the years, a small group of regular contributors have become our special friends. Joseph R. Charnetski and Bob & David Snader have been in this category the longest; more recently, Lucy S. Terrier has become a part of the family.

Besides the staff at TAB BOOKS, three people made special contributions to the production of this book: Pat Baker of Penn Technical Institute and Reg Ney of Duquesne University, who lent the special equipment with which the program listings were made; and Calvin Demmon, who screened every tip for Plus/4 compatibility.

Finally, but most importantly, the greatest contribution has come from a large, remote and all-but-unknowable presence that touches each of our lives. Seldom praised by those it has served, this frequently maligned entity deserves our heartfelt gratitude. So join me in this final acknowledgment: Thank you, Commodore Business Machines, for making those wonderful computers.

Louis F. Sander
Pittsburgh, Pennsylvania
April, 1989

Introduction

This book is for everyone who owns a Commodore computer. Whether you have a 64, 128, Plus/4 or any other Commodore system, it has something of value for you. Chances are, it has hundreds and hundreds of somethings.

The book is an orderly compilation of every item that has appeared in *Commodore Magazine's* popular "Tips & Tricks" column, from its beginning in 1986 through the issue of December 1988, with many additions of new and important material.

Every page contains practical items that will amuse you, teach you, or help you make use of your Commodore. Every item has been used, tested and reused by computerists in many countries of the world.

There are almost 600 Tips & Tricks in all, over 150 of which include programs. Thirty of the most powerful programs have machine language subroutines, each of which is disassembled in the chapter on Machine Language.

Every tip was contributed by a computerist just like you. In soliciting tips, I seek hardware and programming hints, short programs or subroutines, ideas for a better computer room, and similar items of value. Originality counts, but is not an absolute requirement. What's important are usefulness, simplicity, and ease of implementation.

I get what I seek, and each month the best items are published in *Commodore Magazine*. In putting them together for this book, I've rewritten, corrected and improved the previously published items, and I've added plenty of new ones.

These "Tips & Tricks" cover every subject from simple computer room hints through advanced machine language programming, and they include material for all levels of interest and expertise. So whether you just bought a system today, or whether you've had one longer than I have, this book has something for you. Chances are, it has hundreds and hundreds of somethings.

How To Use This Book

You can read this book from cover to cover, or you can use it as a reference source. If you're interested in a particular subject, it's easy to find what you're looking for.

Each chapter heading identifies the primary subject of the tips in that chapter. Within each chapter, tips appear in an appropriate logical order, usually from simple to more difficult, or from general to more specific. Because each tip has a descriptive name, you can easily zero in on your topic.

Most chapters end with a list of related tips in other chapters, which serves as a useful cross-reference.



Tip Dingbats

To make it easy to find the tips and tricks that apply to your computer, the following system of dingbats is used in the tip titles:



Tip for the Plus/4 computer



Tip for the Commodore 64 computer, or the Commodore 128 in 64 mode.



Tip for the Commodore 128 computer

Tip Names

Most tips apply to both the 64 and the 128. Many also apply to other Commodore computers, and even to computers of different makes. That's a bewildering range of possibilities, so I've tried to help you by naming the tips appropriately, as well as by using the dingbats.

If a tip applies only to the 64 or 128, that number appears in its name. If a tip name lacks the telltale 64 or 128, it can be used on either computer, and possibly elsewhere as well. This convention always holds true when the tip includes a program; to the best of my ability, it holds true with the others also.

Chapter 19 "The BASIC Programming Language" is an exception to the rule. That chapter is sequenced by BASIC keywords, and it just wasn't practical to specify which keyword applies to what computer. If you own the computer, you know which keywords are yours.

Program Listings

Because Commodore's screen graphics can be confusing when typing in programs, a special "program listing program" has been used to create all BASIC listings: it converts those graphics into descriptive

English words. You'll find the details in Chapter 2, under "Keystroke Descriptions in Listings." The same keystroke descriptions have been used in the text itself.

The first line of each listing includes the name of its associated tip, plus the name of the author if there's room.

Because of an extensive testing process, "Tips & Tricks" listings are unlikely to contain any errors. If you can't get a program to work, you've probably typed something incorrectly. (Dozens of people write me each year about supposedly defective programs; almost every one has made an error in typing).

If you think you've discovered an error, check and recheck your typing, using the hints from the chapter on debugging. If you fail to find the problem, mail me a complete program listing, a full description of your symptoms, and a self-addressed stamped envelope. I'll help you if I can.

The sure way to eliminate typing errors is to purchase this book's companion disk. It contains every program in this volume, plus a number of unadvertised bonuses. See the back of the book for an order blank.

Submitting Your Own Tips & Tricks

If you have some tips and tricks of your own, you might like to submit them for publication in *Commodore Magazine*. In addition to a modest payment, successful contributors get longstanding satisfaction and momentary worldwide fame. (In this book, tips that have no attribution are my own—no sense in repeating my name and hometown over and over.)

Put each tip on a separate piece of paper that includes your name and address. Put all your tips in one envelope, and mail it to:

*Louis F. Sander
Post Office Box 101011
Pittsburgh, PA 15237*

If your trick is accepted for publication, you'll get a check for \$5-\$50. Your check will be mailed when the trick is printed, which is never less than three months from the time I get your letter. Due to the large volume of mail, I can't acknowledge receipt of your trick, and I can't return unused tricks to you.

I do read every submission, but I give closest attention to those that are printed or typed, double spaced, on 8½ x 11 paper. If your trick includes a program, it's best to send it on a disk. I can return your disk only if you include postage, packaging and a mailing label. I donate unused disks to my user group.

Learning

“Learning is not attained by chance. It must be sought for with ardor and attended to with diligence.”

Abigail Adams
Letter to John Quincy Adams
May 8, 1780

This book begins with tips about learning, because learning is the most important computer subject. Learning is a central theme of working with computers, and much of the fun of using them comes from the learning experiences they provide.

Whatever your computing goals, you can reach them more easily if you follow this chapter’s advice.



Resources for Learning

My computer knowledge has come a long way since I bought my first Commodore PET in 1979. Here's my personal advice for increasing your own knowledge of computers and computing.

First, remember that computer knowledge isn't very hard to acquire. There's a huge amount to know, but it breaks down into tiny individual pieces, each of which can be understood with a modest amount of effort; and once you've learned enough of these elementary pieces, the others will come rather easily. Mostly, it's a matter of how much time you put into learning.

Second, get plenty of hands-on time. There's no substitute for *doing* with a computer, and no part of learning has the potential to be so much fun. As you're enjoying your computer, you're also always learning. You'll be surprised how quickly your hours add up, and you'll be amazed how your understanding multiplies.

Third, always read the manual. Refer to it often, and don't be afraid to write in it. Use a marker to highlight the important parts, and use a pen to make notes in the margins. You'll be surprised how often you return to them. It's also a good idea to put tabs on each important chapter.

The fourth way to increase your computer knowledge is by finding and befriending a teacher. Many schools offer computer literacy courses these days, and plenty of schools offer programming courses; if you stick with the elementary or non-credit courses, you'll find them rather inexpensive.

Every course has a teacher, and every teacher is a potential source of wonderful computer knowledge. Get to know your teacher. Let the teacher know of your interest in the subject. You'll be amazed at how much you will learn.

Teachers are also found in user groups, where they're likely just to be called "friends." Computer friends were my most important resource in the early days; without them, I'd never have gotten past the beginner stage. Only one of them, Vic Pitre, was a genuine expert, but each of them knew something I didn't; and every time one of them would teach me something, we'd both be inspired to learn something else.

So there is Sander's advice on learning about computing: approach the matter with confidence, get plenty of hands-on time, always read your manuals, and find a teacher and a friend.

Louis F. Sander
Pittsburgh, Pennsylvania



Reader's Guide to Computer Literature

Written material is extremely important for learning about computers, and if you aren't an active reader, your computer life will be limited.

Computer literature comes in many varieties, all of which are important to the learner. When you read any of it, remember that every word counts. (It's just the opposite of *People* magazine and the like, where it's perfectly acceptable to skim).

Magazines are an important class of computer literature. My recommendation is to subscribe to one of them and to browse the others at the newsstand. When your browsing uncovers something interesting, buy that issue and take it home.

If you follow that advice, you'll find that your subscription gives you a monthly exposure to new information at a very reasonable price, while your browsing gives you broad exposure to the field.

Speaking of magazines, it's a good idea to keep a copy of any back issue right beside your computer. Go through it every week or so, and you'll be amazed how much you learn with each reading. After six months or a year, you'll understand every word in the magazine; when that happens, you can consider yourself an expert.

When looking at computer books, remember that they're organized into chapters, subchapters and so on. Study the table of contents until you're familiar with the general organization of the book, then dig into the chapters that interest you. Reread each section until you understand it.

A programmer's reference guide is vital if you want to learn to program, because it has specific descriptions of each command and its variations. Mine is tabbed and highlighted to death, with plenty of notes in the margins.

If you want to learn to program, however, the programmer's reference guide is not the book to read. If you want to learn to program, get a book that teaches people to program. These books are available at any bookstore, and many of them are very good. They have titles like "Learning to Program in BASIC." Look for one with sample programs and programming assignments, because when you work them out, you learn. The book I learned from is "BASIC and the Personal Computer" by Dwyer and Critchfield. It wasn't oriented toward Commodore BASIC, but it did the trick for me.

*Mary Lee McCafferty
Butler, Pennsylvania*



Read It and (Don't) Weep

Sometimes you'll be reading a computer article or book, and it seems like you don't understand a single word. If that happens, read the rest of it anyway, trying to understand as much as you can. If you read the same thing again after a week or so, you'll be surprised to see how much your understanding has improved; and if you wait another week and read it a third time, you'll find you're becoming an expert.

What you experience is an example of a common but little known fact of computer literature: repeated readings, spaced by days or weeks, dramatically enhance your comprehension.

You can take advantage of this phenomenon if you own a stack of old computer magazines. Look into some of them and you'll be amazed how many really great articles they have. You'll wonder why you didn't see them the first time through. Well, the magazines haven't changed since you set them aside—but you've changed, by accumulating technical knowledge; and that new knowledge, applied to the old magazine articles, turned them from discards into gems.

*Lucy S. Terrier
Alton, Illinois*



Two Books Are Better Than One

When you get a programmer's reference guide, don't throw away the user's guide that came with your computer. Use it to look at the charts in the appendixes while reading the bigger book.

*James F. Ward III
La Grange, Georgia*



Go to the Meetings!

One of the best ways to increase your knowledge is by attending meetings and conventions where there are other people with interests similar to yours. Most cities have at least one Commodore user group, and attending their meetings can put you in touch with some valuable computer friends.

Even more beneficial are the large Commodore conventions conducted by such groups as WCCA and TPUG (West Coast Commodore Association and Toronto PET Users Group). Manufacturers, dealers, publishers and software houses exhibit at these shows, and at their booths you can see their latest products and talk with people who are really in the know. Usually there are educational lectures as well, where

you can learn about specific topics that interest you, and ask questions that only the experts can answer.

One of the best parts of these shows is the chance to meet the people whose work you've seen in print or online. At one show, in one day, I got to speak with Louis F. Sander, Jim Butterfield, Len Lindsay, Louis Wallace and several others whose articles and programs I've used and adored. I also met online personalities like Quantum Link's MISS CHRIS and BILL P1, and other notables like Commodore Magazine's Jim Gracely and Commodore's Pete Baczor. All in all, attending that convention was one of the most exciting things I've ever done, and I recommend such shows highly to everyone.

*Becky Cassell
Glendale, California*



Finding Fellow Users

For those of you who live in small towns and have no local user groups to fall back on, I suggest running an ad in your local paper. Simply mention the type of computer you own, and that you are interested in exchanging ideas and programs. (Public domain programs and ones you created yourself, of course).

After the ad appears a few times, you're bound to find someone who owns a computer like yours. And who knows...maybe you'll start your own user group!

*Neal J. Pynenberg
Kaukauna, Wisconsin
(Pop. 11,310)*



A Place to Start Your User Group

I'll never forget our first successful meeting at the local Pizza Hut—we loved it and so did the restaurant. Restaurants of all kinds welcome any group of people, especially if they buy food and if they come when business is slow. This is good to keep in mind if you can't find a regular meeting place. You can start out at the restaurant, then move when a more traditional place becomes available.

You can approach the restaurant in advance and ask them what nights they are slowest. They'll probably love to have you. When the meeting rolls around, be sure that everybody orders something, and keep the public domain library away from the food and drink!

*A. M. Johnson, Jr.
Cary, North Carolina*



Finding User Group Meeting Rooms

When choosing a place to meet, you should consider other factors than just the size and cost of the room. Many groups meet in schools or in the members' homes; both places are economical, but they don't have much walk-in traffic, so they're not likely to attract new walk-in members. Neither do they have anything to amuse the spouses, parents or other family members who often drive the members to the meeting.

We've found that shopping malls and public libraries are very good places to have our meetings. Both places usually have meeting rooms that are open to the public, and if they aren't free, their rates are very low. Both places also have a wide range of attraction for family members, and plenty of things to amuse them while you are at your meeting. If you put up signs, you'll certainly attract some new members from the crowds at the mall or the library. The easy-to-find, non-threatening aspect of such places makes them very attractive to those who'd like to join but suffer from some sort of reluctance.

*Pittsburgh Commodore Group
Pittsburgh, Pennsylvania*



The Ultimate Tip

This hint is about graphics, sprites, music, mathematics, programming, hardware and software. It is compatible with any Commodore computer, and is helpful for a great variety of peripherals and commercial software packages. Here it is: Never miss any issue of *Commodore Magazine!*

*Luis Padua
Mexico City, Mexico*

Related Tips in Other Chapters

If your learning goals include programming, you should investigate Chapter 18, "Learning to Program," plus the other chapters to which it will lead you.

Programs from Books and Magazines

“I cannot live without books.”

Thomas Jefferson
Letter to John Adams
June 10, 1815

We Commodore owners are lucky. Because we number close to ten million, there are hundreds of books, magazines and newsletters that are published for us each year. Most of them contain printed programs that we can type, save, use and modify.

This chapter shows how to work more efficiently with those programs, over 150 of which you are holding in your hand right now.



Typing In Programs

Most people find great value in the programs published in books and magazines. If you like to type these programs into your computer, here's a compendium of hints to make the work easier and more rewarding.

- These programs are not only useful on their own merits, but they can help you learn about programming. As you type them in, you absorb the vocabulary and syntax of BASIC, as well as the techniques for putting them together to make useful programs. In fact, the more attention you pay to what you're typing, the more you learn about your computer.

- Typing is much easier and more accurate if a friend reads the lines to you as you type them. If your friend is into computers, you can both learn a lot from the process. If you have a voice synthesizer, you can use it to read you the programs.

- In the absence of a friend, you can read the material into a tape recorder, then play it back as you type. It's useful to have a footswitch for your recorder, so you can stop it between lines or phrases.

- There are many commercially available stands and other devices for holding the magazine as you type from it. Some even have magnifying glasses to make it easier to see your material. Avoid the ones that use magnets, because they can be hazardous to disks or tapes.

- You can use a copy machine to make copies of your programs before you type them in; you can mark or mutilate these copies at will, preserving the value of your magazine. If your copy machine has an enlarging feature, you can use it to make it easier to work with your copies.

- Some people use a felt-tipped highlighter to cross off lines as they type them, while others highlight every second or third line before starting to type. Scotch Post-It notes also make nice movable markers to keep track of your place on the page.

- You can save typing time by eliminating the REM statements in the program. They help in your understanding of the program's flow, but they can take a long time to type. You can often do without the REMs in your version of the program, because you still have them in your original printed copy. If a line contains a REM comment and nothing else, it's a good idea to retain the line number and the REM keyword, even if you eliminate the rest of the line.

This protects you from errors if the programmer has used that line as a target for a GOTO or GOSUB.

- When typing long programs, use the so-called checksum programs made available by the magazine. (You'll find *Commodore Magazine's* printed in every issue, under "Magazine Entry Programs"). By presenting a unique code for each line, they let the computer determine whether the line has been typed correctly. If you use them, there's no need to proofread your work. I haven't used checksums in this book, because most of these programs are short and simple, making proofreading easier than using a checksum program.

- If you don't use the checksum programs, there are many techniques for easier proofreading. The easiest is to list your work on the printer, so you can use the listing rather than the screen from which to proofread. If you use the screen, proofread from the top line, then scroll it up off the screen when you're finished. This way, the line you're working on is always at the top of the screen.

- Because it's very easy to leave out a parenthesis, always count the parentheses in the lines that give you errors. If the total is an odd number, you know you've left one of them out.

- When proofreading DATA statements, always look for extra commas, and for spaces or periods in place of commas.

- Save your work periodically. Nothing is as frustrating as spending hours typing in a program and having something erase it before you save it. Wise computerists save their work several times each hour, sometimes as often as every few minutes. Unwise computerists eventually learn why the wise ones do it.



Program Typing Tip

Because I can never remember the keys used for some of the graphic symbols in type-in programs from different magazines, I cut out the graphic symbol key found in the back of the magazine, and use it as my place marker while typing. It serves as a ready reference for symbols, and it can also be used as a bookmark!

*Todd Moore
Address Unknown*



Magazine Placekeeper

When typing in programs from a magazine, use a reader response card to mark the line that you are typing. Even better yet, use one of

the subscription cards that fall out whenever you hold the magazine. You'll be finding a valuable use for an annoying and unappreciated object.

*Jeff Johnson
Allensville, Pennsylvania*



Read It to the End

I recently typed in the "Disk Signer" program that appears in Chapter 31 of this book. It looked like a neat program, and it was, except for one problem: some of my other programs didn't work after I used the "Disk Signer"!

When I went back to the program's description, I noticed something I had missed the first time around—a specific warning that the "Disk Signer" needed to be run before any other programs were put onto the disk.

This or something similar could happen with many other programs. To avoid it, be sure to read and reread the material that accompanies any computer program.

*Tom Denning
Daggett, Michigan*



Label the Programs You Type

I have a very large collection of programs from the "Tips and Tricks" column and from computer magazines in general. Almost always, the programs had an accompanying article that explained them or gave useful information about their use. Unfortunately, it's very difficult to remember where I got most of the programs, so I'm out of luck when I want to go back to the original source.

To get around the problem, I now record the source of every program whenever I type it in. I put a REM statement in an unused line number at the very start of each program. The REM consists of the magazine name, the month, the year, and the page number of the original article.

Like most such useful habits, I wish I'd started doing this years ago.

*Don Halloran
Bloomfield, New Jersey*



Corrections to Published Programs

From time to time, every magazine prints corrections or improvements to programs it has previously published. Often they appear

as letters to the editor. Whenever I see one of these, I find the original article and make note of the correction right on the original pages. Why? Because one day I might decide to type that program, and I don't want to miss the correction or improvement.

I make these notes religiously, even if the original program didn't interest me. I never know if it will interest someone who borrows my magazine, and I do know that my own interests change greatly over time. In fact, when I dig out a back issue to transcribe one of these notes, I often find that the program it refers to is very much of interest, even though it wasn't so before. (After all, the best programs inspire the most letters to the magazines).

Gregory Mead
Kent, Ohio



Keystroke Descriptions in Listings

All this book's BASIC program listings were created by a special "listing program," that converts the special Commodore graphics characters to an alphabetic form that's easier to read and type. We've used the same listing program that *Commodore Magazine* uses; if you're more familiar with listings from another magazine, it won't be too difficult to make the transition to Commodore's style. Here's the key to typing in our listings:

When you see a word inside brackets, such as [DOWN], that word represents a keystroke or series of keystrokes. Figure 2-1 shows all these bracketed words, along with their corresponding keystrokes. As an example, you can see that the word [DOWN] directs you to press the Cursor Down key.

If multiple keystrokes are required, the number of keystrokes follows the bracketed word. For example, [DOWN4] means to press the Cursor Down key four times.

If one set of brackets contains two or more words separated by commas, enter the keystrokes directly after one another, without entering the commas. For example, [DOWN,RIGHT2] means to press the Cursor Down key once, and then the Cursor Right key two times.

Sometimes a program requires the simultaneous pressing of two keys, one of which is either the Shift, Control or Commodore key. In these cases, the keystroke appears in brackets as SHFT, CTRL or CMDR, followed by the representation for the other key. To give an illustration, [SHFT E] would be entered by holding down the Shift key and pressing the letter E.

Abbreviation	Key To Press
^	Up Arrow
[BLACK]	[CTRL 1]
[BLUE]	[CTRL 7]
[BROWN]	[CMDR 2]
[CLEAR]	Shifted CLR/HOME
[CMDR]	Commodore key
[CTRL]	Control key
[CYAN]	[CTRL 4]
[DEL]	Unshifted INST/DEL
[DOWN]	Cursor Down
[F1]-[F8]	F1 through F8
[GRAY1]	[CMDR 4]
[GRAY2]	[CMDR 5]
[GRAY3]	[CMDR 8]
[GREEN]	[CTRL 6]
[HOME]	Unshifted CLR/HOME
[INST]	Shifted INST/DEL
[L. BLUE]	[CMDR 7]
[L. GREEN]	[CMDR 6]
[L. RED]	[CMDR 3]
[LEFT]	Cursor Left
[ORANGE]	[CMDR 1]
[POUND]	English Pound
[PURPLE]	[CTRL 5]
[RED]	[CTRL 3]
[RIGHT]	Cursor Right
[RVOFF]	[CTRL 0]
[RUN]	Shifted RUN/STOP
[RVS]	[CTRL 9]
[SHFT]	Shift key
[SHFT^]	Pi symbol
[SPACE]	Spacebar
[TAB]	TAB key
[UP]	Cursor Up
[WHITE]	[CTRL 2]
[YELLOW]	[CTRL 8]

Fig. 2-1. Keystroke abbreviations used in this book.

As with other abbreviations, a number following the letter tells you how many times to type the letter. [SHFT A4,CMDR B3] would mean to press the Shift key while pressing the letter A four times, then to press the Commodore key while pressing the letter B three times.

One final point—the Commodore character set includes an Up Arrow character, used in BASIC to indicate exponentiation. (The Up Arrow key is located next to the Restore key). In most non-Commodore computer equipment, including the Diablo printer used to produce the program listings, the Up Arrow is represented by a caret, or ^ . So when you see a caret in the listings, remember that it represents the Up Arrow key on your Commodore.



Commodore Magazine's Magazine Entry Programs

Each issue of *Commodore Magazine* includes two “Magazine Entry Programs” on its final pages—one for the 64 and another for the 128. By automatically detecting typing mistakes, the magazine entry programs make it easier to type the other programs that are printed in the magazine.

In addition to appearing in *Commodore Magazine* each month, the magazine entry programs are available on Quantum Link (see Chapter 13) and from many user groups and bulletin boards. They are also on the disk that's available to accompany this book (see page 391).

When a printed program is set up for use with a magazine entry program, each of its lines ends with an apostrophe followed by a checksum of four letters. (Due to the shortness and simplicity of the programs in “Tips & Tricks”, we don't provide the checksums here).

When you load and run one of the “Magazine Entry” programs, it puts a machine language program in memory. You then execute a SYS statement to activate the machine language program, and a NEW command to remove the magazine entry program itself.

Once the SYS has been executed, the invisible machine language program carefully checks each BASIC line that you enter. It gives audible and visible alarms if the BASIC line lacks the proper checksum. The usual cause of this, of course, is a typing error in the BASIC line itself.

There are ways to disable the automatic checking, and other ways to reactivate it once disabled. Complete operating instructions are printed in the magazine each month.

All in all, the system is simple, accurate and useful. If you haven't tried it yet, you ought to give it a whirl.



Magazine Entry Program Hints

Most programs in *Commodore Magazine* include an apostrophe and four letters at the end of every line. You should not type those characters unless you're using the Magazine Entry Programs described in the back of every issue. The Magazine Entry Programs are used to ensure accurate typing, and they treat the five extra characters as a sort of checksum. The Magazine Entry Programs delete the extra characters from the lines you type, so they don't end up in your program. If you don't use the magazine entry program, but do type the extra characters, they are made a part of the program you're typing, where they do nothing but cause syntax errors.

When typing programs on my 128, I use the "Magazine Entry Program-128" for both 128- and 64-mode programs. I redefine the F7 key to print an apostrophe, which relieves me from making the awkward [SHIFT 7] keystroke. Also, because most of the checksums for DATA statements start with the letter B, I redefine the F5 key as apostrophe-B, which further eliminates keystrokes.

John B. Boniger
Rock Island, Illinois



Magazine Entry Program Bypass

These programs are wonderful aids to accurate typing. But sometimes I want to customize a line of a program as I type it, and the Magazine Entry Program doesn't want to let me.

I've found that putting a space before the line number will allow any line to be entered without a run through the proofreader. So when I want to customize a line, I put a space before its number. If the line is a complex one, I enter it first without the leading space, so the proofreader will check it for accuracy. Once the line is in memory, I use the screen editor to delete the checksum and make my customization. Then I insert a space before the line number, pressing Return to enter the modified line.

Putting a space before the line number is also useful when a line is so short it doesn't need to be checked. Many such lines consist of just the keyword RETURN or NEXT.

One last thing—never, never, never run a program that you just typed in without saving it first. Errors in the program can wipe it out of memory or lock up your computer; if the program has been saved first, you can reload it and work on correcting the errors.

Jeff Fox
Los Angeles, California



Belated Use of the 128 Magazine Entry Program

If you've used your 128 to type a Commodore Magazine program without using the magazine entry program, it's not too late to gain the benefits of automatic proofreading. Just redefine the F1 key by executing

```
KEY 1, CHR$(27) + "K"
```

then list a section of the program. For each line, type F1 and the checksum. The checksum will be appended to the end of the line, and when you press Return, the magazine entry program will do its thing.

```
IMD Walrus  
Quantum Link
```



Magazine Entry Program Improvements

You can modify your magazine entry programs to eliminate the need for remembering the NEW and SYS commands. For the 64 version, just remove the END statement from line 120, then add these new lines:

```
130 PRINT "[CLEAR,DOWN4] NEW"  
140 PRINT "[DOWN2] SYS 49152"  
150 POKE 631,13 : POKE 632,13 : POKE 634,13 : POKE 198,3 : END
```

For the 128 version, remove the END statement from line 120, then add the following lines:

```
130 PRINT "[CLEAR,DOWN4] NEW"  
140 PRINT "[DOWN2] SYS 4864"  
150 POKE 842,13 : POKE 843,13 : POKE 844,13 : POKE 208,3 : END
```

The new lines use the dynamic keyboard technique to execute the NEW and the proper SYS call.

```
Dale E. Roenneburg  
Brodhead, Wisconsin
```



Magazine Entry Program-128 Speedup

As printed in the back of the magazine, this program takes about 30 seconds to poke its machine language into memory. By adding these two lines, that time is cut in half:

```
7 FAST  
115 SLOW
```

Because the screen is blanked out when the 128 is in FAST mode, lines 10 and 80 can also be deleted.

James Hallman
Owensboro, Kentucky



Modifying the Magazine Entry Programs

The next three items will make your magazine entry programs easier and faster to use.

To be completely equipped, you should set up one disk that contains all the programs you'll need. Here's what should be on it:

- 1) The original "64 Magazine Entry Program," as it appears in the back of *Commodore Magazine*.
- 2) The original "128 Magazine Entry Program," also from the back of the magazine.
- 3) The improved "64 Magazine Entry Program" (the one with lines 1-140 for the 64 added).
- 4) The improved "128 Magazine Entry Program" (with added lines 1-140 for the 128 added).
- 5) The 64 machine language version, made by running #3 above and answering "Y" at the prompt. MEP64.ML will be its name on the disk.
- 6) The 128 machine language version, made by running #4 above and answering with Y at the prompt. MEP128.ML will be its name on the disk.
- 7) The "MEP Autoloader" program from the end of this chapter.

With these seven programs, you'll be able to do anything you'd like with the magazine entry programs, including going back to the originals as printed in the magazine. In most cases, you'll probably just use "MEP Autoloader" which in turn will use MEP64.ML or MEP128.ML, depending on the machine on which it is being run.

These improvements include tips from Larry A. Sexton, Harriman, Indiana; Grzegorz Szmidt, Warszawa, Poland; and Jeffrey A. Seay, Maumelle, Arkansas.



Modified 64 Magazine Entry Program

Figure 2-2 shows lines that are to be added to "Magazine Entry Program-64." To add them, you must first load "Magazine Entry

```

1 REM MODIFIED 64 MAGAZINE ENTRY PROGRAM
2 REM SEE TIPS & TRICKS BOOK
3 :
4 INPUT"SAVE A ML VERSION (Y OR N)";Z$:Z$=LEFT$(Z$,1)
5 IF Z$="Y" OR Z$="[SHFT Y]" THEN Z$="Y"
6 IF Z$="Y" THEN OPEN 1,8,8,"MEP64.ML,P,W"
  :PRINT#1,CHR$(0)+CHR$(192);
100 B=H*16+L:POKE P,B:T=T+B:P=P+1
102 IF Z$="Y" THEN PRINT#1,CHR$(B);
104 GOTO 30
115 CLOSE 1
120 PRINT"[CLEAR,DOWN13]DONE! PRESS <RETURN> TWICE TO ACTIVATE.
130 PRINT"[DOWN]'KILL' DISABLES, 'SYS49152' REACTIVATES."
140 PRINT"[HOME,DOWN3]NEW[HOME,DOWN6]SYS49152[HOME]":END

```

Fig. 2-2.

Program-64' into memory, either by typing it or loading it from disk.

Then type lines 1-140 from Fig. 2-2, which will be added to the original program. (Actually, line 100 replaces a line in the original program.) Don't overlook the semicolons at the end of lines 6 and 102.

Notice that the program now starts with a line identifying it as an improved version. Save it, giving it a name that will differentiate it from the original.

The improved program automates your NEW and SYS by printing them onto the screen, where you can execute them by pressing the Return key. This reminds you to execute these two important commands, and saves some typing when you do it.

Even better, it lets you save the machine language subprogram directly, rather than in its BASIC form. This makes for faster loading every time you use the program. (Loading time is cut from 80 seconds down to five!) To activate this feature, just answer "Y" when asked if you want to save an ML version.

Various Contributors



Modified 128 Magazine Entry Program

The lines in Fig. 2-3 are to be added to "Magazine Entry Program-128". To add them, you must first load "Magazine Entry Program-128" into memory, either by typing it or loading it from disk.

Then add lines 1-140 to the program, by typing them and pressing Return at the end of each. Notice that line 120 replaces a line in the original program.

Chapter 2

```
1 REM MODIFIED 128 MAGAZINE ENTRY PROG
2 REM SEE TIPS & TRICKS BOOK
3 :
4 INPUT"SAVE A ML VERSION (Y OR N)";Z$:Z$=LEFT$(Z$,1)
  :IF Z$="[SHFT Y]" THEN Z$="Y"
115 IF Z$="Y" THEN BSAVE"MEP128.ML",B0,P4864 TO P5546
120 PRINT"[CLEAR]MAGAZINE ENTRY: SYS4864=ON[SPACE2]SYS4867=OFF"
125 W=39:IF RWINDOW(2)=80 THEN W=79
130 WINDOW 0,1,W,24
135 PRINT"[DOWN10]DONE! PRESS <RETURN> TWICE TO ACTIVATE.
140 PRINT"[HOME,DOWN3]NEW[HOME,DOWN6]SYS4864[HOME]":END
```

Fig. 2-3.

Observe that the amended program now starts with a line identifying it as an improved version. Save it, giving it a name that will differentiate it from the original.

The improved program puts a line of instructions in a window at the top of the screen, where it stays visible as you enter your other programs. It also automates your NEW and SYS by printing them onto the screen, where you can execute them by pressing the Return key. This reminds you to execute these two important commands, and saves some typing when you do it.

Even better, the improved program lets you save the machine language subprogram directly, rather than in its BASIC form. This makes for faster loading every time you use the program. To activate this feature, just answer with Y when asked if you want to save an ML version.

Various Contributors



MEP Autoloader

The program in Fig. 2-4 automatically manages the machine language programs that were optionally created by the two improved programs above. It detects whether it's running on a 64 or 128, then loads and executes the appropriate machine language code. It even displays the commands for disabling and reactivating the magazine entry programs.

To use "MEP Autoloader," be sure the machine language version of the proper magazine entry program is on the disk in your drive. Then load and run "MEP Autoloader" and everything will take care of itself.

Various Contributors

```
100 REM MEP AUTOLOADER - SANDER
110 REM AUTOLOADER FOR ML VERSIONS
120 REM OF COMMODORE MAGAZINE
130 REM MAGAZINE ENTRY PROGRAM
140 PRINT "[CLEAR,DOWN]WORKING...":K$="KILL":P$="MEP64.ML":R=49152
   :REM 64
150 IF PEEK(65534)=23 THEN K$="SYS 4867":P$="MEP128.ML":R=4864
   :REM 128
160 IF A=0 THEN A=1:LOAD P$,8,1
170 PRINT "[CLEAR,DOWN]COMMODORE MAGAZINE'S MAGAZINE ENTRY"
180 PRINT "PROGRAM IS NOW ACTIVE."
190 PRINT "[DOWN]ENTER ';'K$;"' TO DISABLE IT."
200 PRINT "[DOWN]ENTER 'SYS";R"[LEFT]" TO REACTIVATE IT."
210 PRINT "[DOWN]COMPLETE INFORMATION ON HOW TO USE THE"
220 PRINT "MAGAZINE ENTRY PROGRAM CAN BE FOUND IN"
230 PRINT "THE BACK OF ANY ISSUE OF THE MAGAZINE.[DOWN]"
240 SYS R : NEW
```

Fig. 2-4.

Related Tips in Other Chapters

These chapters relate in a general way to *Programs from Books and Magazines*:

Chapter 5	Keys and Keyboarding
Chapter 18	Learning To Program
Chapter 27	Debubbing

Saving Money

“Whatever you have, spend less.”

Samuel Johnson
December 7, 1782

Let's face it, few of us are wealthy. When there's a new piece of software to be bought, or even a new ribbon for the printer, we don't always have the money at our fingertips. In fact, we often don't have it anywhere. Here are some ways to spend less of that oh-so-precious commodity.



Finding Computer Bargains

In late summer and early fall, many amateur radio clubs hold “hamfests” where radio and other equipment is sold in a flea market atmosphere. Luckily for us, most computerized hams use Commodore equipment, so their hamfests can be very good hunting grounds.

At the TSRAC (Wheeling, West Virginia) hamfest last year, I saw an SX 64 for \$200, several 64’s for under \$100, a 1541 for \$85, and a nice monochrome monitor for \$35. I also saw several semi-commercial vendors with good prices on disks, computer paper and the like.

Believe it or not, another excellent source of bargain equipment is the non-Commodore computer store. If they have any Commodore-compatible equipment, they probably don’t even know how to hook it up. Because your potential treasure is useless to them, they’ll probably price it accordingly.

At a recent “warehouse sale” of shopworn and trade-in equipment, run by an IBM-oriented dealer, I saw three Commodore-compatible printers for \$25 each! Because they had the serial bus DIN plug and no Centronics-type connector, nobody in the store knew what to do with them.

Used monochrome monitors were in the same store for \$25 and \$35, and several pieces of Commodore software for \$2 each!

*Louis F. Sander, W3BOA
Pittsburgh, Pennsylvania*



Free Paper, Free Disks, Free Equipment!

Tons of used computer paper are thrown away each day by companies of all description, and every sheet of it has one unused side just waiting to be run through your printer. You can use this paper for drafts, program listings and the like, and you can usually get it free. Just ask the person who’s throwing it away, and you’ll probably get permission.

If you find a stack that’s wider than your printer, but good in every other respect, you can use it if you have access to a band saw and a printer with friction feed. The band saw will neatly and easily cut through a two- or three-inch stack of computer paper. The friction feed printer won’t mind a diet of sawed-up paper.

Companies that are moving their offices can also be a good source of disks, and even equipment. They often change their computer systems when they move, and they often discard the old stuff. A friend of mine

got two perfectly good monitors this way, plus a ton of 5 ¼-inch diskettes and lots of other useful stuff. Another friend got a boxful of RS232 cables. Keep your eyes and ears open for treasures like these, but always ask permission before taking anything.

*A. M. Johnson, Jr.
Cary, North Carolina*



Enhancing Your Warranty

Commodore products and many products for Commodore systems come with a 90-day warranty from the manufacturer. In many cases, you can double the warranty to 180 days if you pay for the product with an American Express credit card. For details, see your cardmember literature on American Express Buyer's Assurance, or write that department at Torrance, CA 90504.

*D. L. Jassby
Princeton, New Jersey*



Tax Tips

If you itemize deductions at tax time, there may be several ways to deduct the cost of your computer equipment, supplies and software purchases. Some of the major ones are discussed below, but needless to say, none are guaranteed to be effective in your case. Your best bet is to consult with a qualified tax expert or attorney.

Use in Business Particularly if you are self-employed, using your computer for business purposes will generally produce tax benefits. The cost of your equipment can usually be depreciated, and disks, paper and other supplies can be deducted as business expenses. The one major pitfall is that your computer must be used solely for business purposes. If the IRS discovers that you have been playing ZAXXON, you may be out of luck.

Use for Tax Preparation At least a portion of your computer costs may be deductible if related to tax return preparation or the maintenance of tax records. Commercial tax accounting software is a good example. Also, a database used primarily to store tax information will likely be allowed as a deduction.

Computer-related Income Efforts to generate income by developing commercial software or selling articles to computer magazines may give rise to business-type tax benefits. A major factor is whether you have in fact generated income—even if less than your

reported expenses in a given year. Consequently, selling an occasional program, magazine article or computer hint may make a big difference in keeping the tax man at bay. If you can show a profit in two years out of five, your efforts are presumed to be for profit rather than recreation.

*Kent M. Bridwell
Los Angeles, California*



Saving Money on Printed Matter

You don't have to own a book to read it. Progressive public libraries have an up-to-date collection of computer books, and subscriptions to at least a few computer magazines. If you don't find what you want on the shelves, ask one of the reference librarians.

If the library doesn't have what you're interested in, be sure to let the librarian know about it. She'll appreciate your advice, and if you convince her of the benefits to her patrons, she may even purchase the specific items you're looking for.

Many user groups also have lending libraries of books and magazines. Very often back issues are given away at the meetings.

And don't forget your local bookstore. If you're polite and don't abuse the privilege, most bookstore managers will let you read to your heart's content. By thoroughly browsing the computer section, you'll learn which books and magazines are of most interest to you, and which would be a waste of hard-earned cash.

*Mary Lee Resnick
Butler, Pennsylvania*



Classified Ads Save Money

Many newspapers have a classified advertising category for computer equipment. In the Pittsburgh Press, for example, classification 483A is Personal Computers, Games & Accessories. On a typical weekday, it has 15 to 20 computer ads, almost always including secondhand Commodore equipment.

*Eugene Harter III
Pittsburgh, Pennsylvania*

Related Tips in Other Chapters

As you might imagine, many tips in other chapters have a money-saving aspect to them. Here they are:

Chapter 3

Chapter 4	Easy Printer Stand Household Items in the Computer Room
Chapter 8	Inexpensive Cleaning Fluid Home-made Fanfold Paper Saver Inexpensive Heat-sensitive Paper Cheap Printer Labels Keep Two Ribbons New Life for Old Ribbons Rejuvenating Cloth Ribbons Ribbon Re-inking
Chapter 10	Datassette Tips
Chapter 11	Disk Buyer's Guide How to Make "Flippy" Disks
Chapter 13	Telecommunications Privacy Online Recording Minimizing Telecommunications Charges
Chapter 16	Cartridge Lifesaver
Chapter 33	Save Your Receipts

The Computer Room

*“Here in a little lonely room
I am master of earth and sea,
And the planets come to me.”*

Arthur Symons
The Loom of Dreams
1900

Some of us spend hundreds of hours in our computer room each year. Others log that much time every month. All of us have the opportunity to create a more comfortable environment for our computers, our guests and ourselves.

Here are some suggestions for lighting and furnishing the most important room in the house, for making it a more efficient workplace, and for making profitable use of the byproducts of personal computing.



Glare-free Lighting

While remodeling my computer room, I discovered the perfect lighting system — glare free, and with adjustable brightness to suit both keyboarding and reading or other close-up work. Best of all, the total system costs under \$30, including even the light bulbs.

My system is perfect for computers sitting on flat tables without overhead shelves or hutches, but it can be adapted to many other situations. It uses indirect lighting in “computer mode”, and direct lighting in “reading mode”.

The raw materials are two adjustable-arm lamps of the kind made by Luxo, Ledu and many others. They mount on a desk or table, and hold a single shaded bulb at the end of an adjustable swing-arm assembly consisting of four rods and four springs.

Lamps like this are available from many different sources—hardware stores, office supply stores, and contemporary furniture stores to mention just a few. The expensive models cost over \$50, but discount stores often have attractive ones for about \$10. The bulbs are usually sold separately.

Choose your lamps based on style and price, but find a model rated to hold a 100-watt bulb. These usually have ceramic sockets, while the lower-rated units have sockets made of plastic. My 100-watt Ledus cost \$12 each, and they’d be at home in the most luxurious office.

Put 100-watt bulbs in your lamps, then mount them behind your monitor, one on the right and one on the left. Mine are about 9 inches to either side, where they don’t interfere with other items on the desk.

The secret to glare-free operation is to use the lamps for indirect lighting. In computer mode, place them higher than your head and aim them toward the wall and ceiling. With a little experimentation, you’ll find a position that gives ample light on your desk and keyboard, with little or no glare on your screen and no light source directly in your vision. Tighten their knobs until the lamps stay where you want them, and stand by for well-lighted glare-free computing.

While 200 watts of indirect light is fine for most keyboarding, it’s not quite enough for serious reading, or for looking closely at equipment. For those applications, turn one or both of your lamps around so they shine directly on the desk, and move the shaded bulbs closer to what you’re working on. Two hundred watts of direct incandescent light, up close and very personal, is more than enough for any computer room task.



Easy Printer Stand

You can make a perfect printer stand from one of the desk trays sold at office supply houses and discount stores. Turn the tray upside down, with its opening toward the rear, and set your printer on top of it. Your paper will fit in the space between the desk and the top of the "printer stand" (really the bottom of the desk tray), and can be fed out the opening and up the back of the printer.

These trays are available in a wide variety of colors, shapes and sizes, so you should be able to find one exactly right for your needs. The better brands come in letter, legal, and data processing sizes; the latter is ideal for wide-carriage printers.

Luke A. Babey
Saddle Brook, New Jersey



Monitor Shelf

Commercial decorator shelf parts, suitably modified, can make a rugged adjustable shelf for your video monitor. The type I use has aluminum brackets that slide into slotted aluminum channels which mount on the wall. Both the channels and the brackets come in a variety of finishes. The most common brand in my area is K & V, which is carried in most hardware stores.

Normally, these shelves are not sturdy enough for something like a monitor, but with simple modifications, they work like a charm.

To make my shelf, I bought one three-foot channel and two 12-inch matching brackets. I cut the channel into two equal pieces and mounted them 12 inches apart on the wall behind my computer table. Simple plastic screw anchors were strong enough for the job.

The wooden shelf itself is the key to a rugged installation. I cut a piece of plywood just wide enough for my monitor to sit on, and an inch deeper than the length of the shelf brackets. For my 12-inch brackets, this was 13 inches deep. Then I cut two notches in the back of the plywood, just the size of the channels on the wall (about $\frac{5}{8}$ -inch wide by $\frac{1}{2}$ -inch deep). They are positioned 12 inches apart, where they fit snugly over the two mounting channels. When the shelf is in place, the notches let the plywood sit flush against the wall, preventing sideways movement at that point.

The tops of the 12-inch brackets have little raised hooks on the front. The wooden shelf usually fits between these hooks and the wall; they prevent it from moving forward and perhaps to the floor. In our heavy-duty installation, we take better advantage of the hooks. Because our

shelf is cut large, it extends about ½-inch forward of the raised hooks. On the bottom of the shelf, I carefully marked the locations of these hooks, then drilled ⅛--inch holes about ¼-inch deep right there. When I installed the shelf, the hooks fit into the holes, and the shelf is not only kept from moving forward, but also from side to side. The arrangement also keeps the brackets from moving sideways, which is a common problem with this type of shelf.

My finished shelf cost less than 10 dollars; it is attractive, adjustable, and solid as a rock. It gives me lots of surprisingly valuable room on my computer table. A similar shelf could do the same for you.

*William F. Sander
Pittsburgh, Pennsylvania*

○△☆ **Lightning Detector**

Remember this tip during thunderstorm season: A radio tuned to a clear channel at the low end of the AM band makes an excellent lightning detector. You can usually hear the characteristic lightning crackle on the radio long before you hear the thunder or see the approaching darkness.

When you hear the static, it's a good time to save your program, shut down and unplug. It's not very likely that lightning will damage anything, but you're better safe than sorry. Don't forget the modem—it's a good idea to unplug it from the telephone line.

*Frank Colaricci
Winter Park, Florida*

○△☆ **Connector Labels**

Like most computers, mine sits on a desk underneath an overhanging shelf, attached to a rat's nest of wires and cables. It's hard to see the back panel at all, let alone well enough to plug in another wire. So when I need to work with the plugs, it's very hard to locate them. I've solved the problem by labeling each connector on the TOP surface of my computer, way at the back, immediately above the connector itself. Now when I need to locate the serial bus or some other connector, I know it's right below its label. Once I have it located so precisely, it's easy to replace it by feel alone.

*Cecil C. Serfent
Chesapeake Beach, Virginia*



Handy Notes

Does your monitor or other equipment sit on a raised shelf in front of you? If you put a piece of masking tape on the front edge of that shelf, you can use it as a place to record POKEs, SYS numbers, and other frequently-used but hard-to-remember information. It's difficult to find a more convenient location for this all-important data.

*Lucy S. Terrier
Alton, Illinois*



SAVE "YOUR MARRIAGE", 8

I share my computer desk with my wife. My Commodore 64 is on one half of the desk, and her writing and sewing area is on the other. This way, my wife and I can be together while doing the things we like.

Since we started this arrangement, I've had more time to do computer things, and she has never complained about my spending too much time with "that machine". It's a very good way to save your marriage or your computer.

*Timothy Sullivan
Brooklyn Park, Minnesota*



Nice Marker

An exceptionally good highlighting marker to have in your computer room is the Stabilo Boss 2, available at most office supply stores. It's Stabilo's refillable model, available in several different colors (my favorites are yellow, orange and green). The Boss 2's shape keeps it from rolling around on the desk, and its chisel tip is exactly the right size for highlighting computer printouts or the listings found in magazines.

*Robert L. Sander
Pittsburgh, Pennsylvania*



Printer Ruler

I've found a special ruler that's invaluable for use with a printer. It's 14 inches long, with four scales marked in inches, lines of text, 10-pitch characters and 12-pitch characters. I use it for layout work, for counting characters in listings, etc., and it's saved me lots of time and aggravation.

Made by the C-Thru Ruler Company, the model WP-101 is available for about \$3.00 at many office supply stores.



Household Items in the Computer Room

Who says computers are a rich man's hobby? You can use many articles from around your house to help you with your computer. For example:

- Use Windex or a similar glass cleaner to clean your monitor screen and the plastic surfaces of your equipment.
- Use a hole puncher as a disk notcher.
- Use masking tape or electrical tape as a disk write-protect tab.
- Use an old shoe box to store your floppy disks.

*Victor R. Heredia
El Paso, Texas*



Inexpensive Cleaning Fluid

When the liquid that came with my disk drive cleaning kit was left open, it evaporated and I couldn't find any replacement fluid. When I looked at the listed ingredients, I found that this fluid was primarily isopropyl alcohol.

Since then I have been using isopropyl alcohol from the drug store. It has 91% purity (compared to 70% in the original fluid) and has no coloring or other additives. It works great in my head cleaning kit, and I also use it to clean the rubber platen on my printer. It's about a dollar a pint at any drugstore. Isopropyl alcohol also makes a good keyboard cleaning fluid.

Q-Link's ClarenceW1 reminds us to be careful when buying this material, because there's a big difference between isopropyl alcohol and isopropyl RUBBING alcohol. The latter usually contains a trace of glycerine to make it less drying to the skin. Unfortunately, the glycerine remains after the alcohol evaporates, and it could be detrimental to your disk drive head or other sensitive parts.

*Joshua Schulman
Valley Stream, New York*



Printer Scrap to Your Notebook

Here's a use for those single sheets of tractor-feed paper. Tear off the left perforation in the normal way, but leave the right one alone. Use scissors to cut into the right perforation at the top and bottom of an area that is to become a "notebook tab," similar to the tabs on divider sheets you can buy in the store. Tear off the perforations above and below your "tab," and you have the world's cheapest notebook divider. By skillful use of the scissors, you can make a complete set of dividers, with tabs arrayed from top to bottom.

*Robert Burnet
Islington, Ontario
Canada*



Printer Scrap to the Microwave

Being a computer widow, I rarely find any interest in what my husband or children do with the computer. However, when I saw my husband disposing of reams of used computer paper, I realized that it could serve as a great liner for my microwave oven.

I just hang the fanfold paper around the perimeter of the oven, using the unremoved tractor holes to hook onto the oven's interior frame. When the paper catches enough splatter, I replace it. It doesn't burn, it doesn't affect the operation of the oven, and best of all, it's free. Who says a home computer can't be useful?

*Harriet Schulman
Valley Stream, New York*

Related Tips in Other Chapters

Much of the material in Chapter 14, "Wiring and Connectors", relates closely to the computer room. These other tips also include pertinent items of interest:

Chapter 6	Monitor Glare Reduction Sight Improver Screen Marking Hint
Chapter 9	Label Your Joysticks
Chapter 13	Telephone Extensions Marking Telephone Cables
Chapter 14	Label That Cable

Keys and Keyboarding

“The used key is always bright.”

Benjamin Franklin
Poor Richard's Almanack
July, 1744

A computer keyboard is the biggest set of controls that most of us will ever operate. Just think what a 128 owner has at this fingertips—92 on/off switches, with four of them (Control, Commodore and two Shifts) altering the function of some, but not all, of the others. Add a 19th century layout carefully designed to slow down the typist, and you see what an awesome control panel that keyboard really is.

It's no wonder that keyboard hints are among the most numerous and useful I receive. Here are two dozen of the best ones.

○△☆ **Slow Listing**

If you know this one, you think everybody knows it. If you don't know it, you think it's a miracle.

To slow down the listing of programs or directories, hold down the CTRL key in 64 mode or the Commodore key in 128 mode or on the Plus/4. Unfortunately, this trick won't work in CP/M mode.

*Sebastian Camberos P.
La Paz, Bolivia*

○△☆ **The "Don't Do It" Key**

How many times have you typed halfway through a line, only to decide that what you've typed so far is garbage? Well you don't have to delete all those characters before you start over again! All you need do is press the "don't do it" key.

Every computer has one of these keys, but it's amazing how many people don't know it. On all Commodore computers except the Amiga, the "don't do it" key is the Shifted Return key. If you press it while the cursor is on a line, the cursor will jump to the start of the next line, but the computer will not process what was on the original line.

The Amiga's "don't do it" key is Control-X.

*Jim Butterfield
Toronto, Ontario
Canada*

△☆ **Easy RUNning**

Here's a one-touch method for running any program in memory on the 64 or 128. Just hold down the left Shift key and press the 2 and 4 keys at the same time. With practice, you can hit all three keys with the fingers of one hand.

*Khoi Luu
Tehachapi, California*

○△☆ **More Easy RUNning**

If you have a program in memory, you can run it very easily by typing any letter or letters (not numbers or other characters), then simultaneously pressing Shift and Run/Stop.

*Greg Heide
Great Falls, Montana*

○△☆ **Typing Large Numbers**

If you need a large number you can often save keystrokes and memory by using the Up Arrow key or exponential notation. It's easier to type 10^6 than it is to type a million, and it's almost as easy to type 10E6.

*Sam Hertzler
Carlisle, Pennsylvania*

○△☆ **Abbreviating Zero**

When a variable is assigned the value zero, such as in the statement $N = 0$, you can replace the zero character by a decimal point! The computer will interpret the decimal point as zero, and will evaluate the expression significantly faster than if you had used the zero.

The trick works in direct mode as well, where it is especially useful in POKE statements. If you want to poke a zero into location 53281, for example, you can type:

```
POKE 53281,.
```

which is a little easier to type than

```
POKE 53281,0
```

You can use this trick wherever a zero is needed by itself, but not when the zero is part of another number, such as 200.

*Itty Greer
Glenshaw, Pennsylvania*

BASIC Abbreviations



Your user manual lists short abbreviations for most of the BASIC keywords. Here's a summary of useful information about them.

- The abbreviations are nothing more than a convenient way of entering statements from the keyboard. They don't save memory, even though they look like they should.
- In addition to saving keystrokes, their greatest use is in squeezing extra keywords into a screen line. For example, the 64's screen editor limits line entries to 80 characters; if you abbreviate keywords, you can get more of them into the line. The principle works on other Commodore machines as well, whether the screen editor limit is 80 characters or something different.
- When you list a line that includes abbreviated keywords, the computer expands them to their fully spelled form. This can make the listed line longer than the screen editor's limit, but the

line will still work perfectly. The thing you can't do is edit a line longer than the screen editor's limit; if changes to such a line are needed, you must enter it again from scratch.

- You can avoid having to retype an abbreviated line when editing, by making up a dummy line. Number the dummy so it will never be executed, and use a quotation mark as the first character after the line number. Then type your regular program statements, including abbreviations. Now list the dummy. Because of the leading quotation mark, it will include the abbreviations rather than their expansions, and will therefore not exceed the screen editor's limits. Change its line number to the one you want the actual line to have, delete the leading quotation mark, then press Return. List the line you've just entered, and observe that the abbreviations have been expanded. If you need to edit this line, list the dummy again, change the line number and delete the leading quote. Then edit the rest of the line and press Return to enter it. Slick, eh?

- It's easier to work with abbreviations if the Shift/Lock is off because the abbreviations then won't contain hard-to-read graphics. You can put your machine into this mode by simultaneously pressing the Shift and Commodore keys.

- The proper abbreviation for PRINT# is P Shift-R. If you try to use ?# the resulting line will look all right, but will give a Syntax Error when executed. Similar restrictions apply to GET# and INPUT#.

☆ **Watch Those 128 Abbreviations!**

One drawback to the 128's BASIC is that some of the keyword abbreviations have been changed. Programmers who have used P Shift-O as an abbreviation for POKE will be surprised at the error message returned in direct mode or at the appearance of the keyword POT in their program listings. Other abbreviations have changed as well, and here's a partial list:

Keyword	64 Abbreviation	64 Abbreviation Gives in 128 Mode	Correct 128 Abbr.
PEEK	p[SHFT E]	PEN	pe[SHFT E]
POKE	p[SHFT O]	POT	po[SHFT K]
READ	r[SHFT E]	RECORD	re[SHFT A]
STOP	s[SHFT T]	STASH	st[SHFT O]
CONT	c[SHFT O]	CONCAT	None
END	e[SHFT N]	ENVELOPE	None
SPC (s[SHFT P]	SPRITE	None

*Charles Lavin
Coral Gables, Florida*

○△☆ **Typing Similar Lines**

Commodore's screen editor makes it very easy to duplicate lines in a program, or to create slightly modified lines. All you have to do is list the line in question, change the line number and anything else that's necessary, then press Return. Both the new line and the old one are now in memory for you to use.

You can use this to your benefit when typing in programs from books or magazines. Many times, entire sections of such programs are virtually identical to one another, and would lend themselves to duplication or semi-duplication by the method described above. Before starting to type a program, you should make a quick search for these sections. Doing this will save you time and will prevent some finger errors. But best of all, it will replace some of typing's tiring drudgery with a more satisfyingly creative method of data entry.

*Walt Latocha
Oak Park, Illinois*

△☆ **Keyboard Curiosity**

Almost everybody knows you can switch between the two Commodore character sets by pressing Shift and the Commodore key at the same time. For an unusual surprise, depress Run/Stop and hold it. Then while it's down, simultaneously press Shift and Commodore.

*Gilbert Figueroa
Hollis, New York*

○△☆ **Keyboard Cleaning**

A quick and effective way to clean between your keys is to use a piece of wide masking tape. Fold it over on itself sideways so about half the piece is sticky, and about half is thicker and folded over. Holding the folded side, run the sticky side down between the keys, where it will collect the dust, dirt and hair that's accumulated there. If you do this every few months, you won't be troubled with intermittent keys.

If you already have some keys that don't work, or that work intermittently, it's not too difficult to clean up their internal workings, if you're a careful worker. (If you've never worked on electronic equipment before, get help from a friend who has.) These instructions

apply to the 64, but the general principles apply to the 128 and all other Commodore machines:

Remove the three screws on the bottom of your computer, and carefully separate the top and bottom shells. Gently separate the connectors to the LED power light and the keyboard. Carefully de-solder the two wires going to the Shift/Lock key, and you're ready to disassemble your keyboard.

Remove all the tiny screws you see on the back of the keyboard assembly, carefully setting them aside in a safe place. When the last screw has been removed, you can lift off the back of the keyboard assembly. At this point, you'll probably see the "gunk" that's giving you your problem. Remove it, then use a soft cloth and some cleaning fluid to wipe off the printed circuit board inside the keyboard. Be careful, gentle and observant, and everything will go well.

When everything is clean and dry, reverse the disassembly process. Test every key on the keyboard, then stand by for years of trouble-free keyboarding.

*Viet Ho
Pasadena, California*



Readable Multicolor Characters

If because of an error or crash you find yourself in multicolor character mode, it may be very difficult to read what you type on the screen. In certain colors, the characters appear as semi-legible ultramodern block letters.

If you change the character color to one of the first eight colors, what you type will be perfectly legible. Use the Control key and a number from one to eight, selecting a color that contrasts with your screen color. White or black are almost always good.

*Victor H. Pitre
Castle Shannon, Pennsylvania*



All about the Stop and Restore Keys

Are you in the dark about these powerful and mysterious keys? Read on and see the light.

The Stop key is on the left side of your keyboard, just above the Commodore key. If you press it while running a BASIC program, it causes an immediate halt, unless the program is waiting for INPUT from the keyboard. (Later on, we'll see how to break out of an INPUT statement.)

As you've probably noticed, the key is labeled Run as well as Stop. That's because when it's shifted, it automatically loads and runs a BASIC program. The particular program it chooses depends on your computer and what you've been doing with it lately. In the 64, the Run key loads and runs the next program found on the Datasette. If you have a 64 but don't use tapes, you won't be able to use Run.

On the 128 and SX 64, the Run key loads and runs a program from the default disk drive, normally drive #8. In most cases that program will be the first one shown in the directory, but that's not always the case. If you've recently saved or loaded a different program, that program may be the one that Run chooses. If you experiment with your own computer, you'll soon know how it works.

The Restore key is in the upper right corner of your 64 or 128 keyboard, just below Inst/Del. To use it, depress Stop, and while that key remains depressed, give Restore a sharp tap, similar to what a woodpecker would do. (Unlike the other keys, Restore is connected to a circuit that detects an instantaneous change in status. If you're too gentle with the Restore key, the circuit can miss your keypress).

Simultaneously pressing Stop and Restore will break out of almost any program, including a BASIC program executing an INPUT statement. In addition, Stop-Restore resets the computer to its power-up configuration for colors, sound and graphics. When you do a Stop-Restore, the normal screen, border and character colors will return, and all sound registers will be reset; all sprites will vanish from the screen. On the 128, the 40/80 DISPLAY key will be read and the appropriate video output will be activated.

Also on the 128, you can escape from an INPUT statement without affecting the screen or sound. Just hold down Stop and press the Enter key on the numeric keypad. (This trick doesn't work with the Return key, which for most other purposes is identical to Enter).

It's possible to alter the actions described above, and many creative programmers have done so. The Stop key can be disabled, and so can the Restore key. Restore can be given an entirely new meaning; and Run can be simulated from within a program. Full coverage of these subjects is beyond our scope at this time, but here are some tidbits for the 64:

- POKE 808,239 disables Stop
- POKE 808,225 disables Stop-Restore
- POKE 808,127 disables Run
- POKE 808,237 undoes any or all the above
- POKE 816,32 causes Run to run what's in memory

On the 128 and Plus/4, the TRAP statement can be used to disable the Stop key, and the programmable function keys can easily LOAD or RUN whatever program you'd like.



Disabling the Stop and Restore Keys

On the 64, POKE 808,225 will disable the Stop and Restore keys, separately or in combination. POKE 808,237 will enable them again.

On the 128, POKE 808,103 disables Stop and Restore, while POKE 808,110 enables them again.

*The Terrier Twins
Alton, Illinois*



Disabling the 64 Restore Key

Sometimes it's nice to disable the Restore key, or to have it do something other than its usual mini-reset. After all, interrupting some programs would be disastrous, or (horrors!) might allow somebody to copy them.

Pressing the Restore key causes program execution to jump to a routine pointed to by a vector in memory locations 792 and 793. Because those locations are in RAM, you can change the vector and thereby change the function of the Restore key.

To cause a system reset when Restore is pressed, do this:

80 POKE 792,226 : POKE 793,252

To undo the change, do this:

90 POKE 792,71 : POKE 793,254

This more general routine can direct execution to anyplace in memory:

100 R = 64738 : HI = INT(R/256) : LO = R - 256*HI :
POKE 792,LO : POKE 793,HI

Set R to the address to which you want a Restore to be directed. Undo the change by executing the code in line 90 above.

*Nolan Whitaker
Jeffersonville, Kentucky*



64 Restore/Reset Disable

Here's a nice one for program protection fans. Using Fig. 5-1, add lines 10-40 to the beginning of your 64 program, and neither the Stop-Restore key combination nor the Reset button will work. Try it.

Mete Kesen
Ankara, Turkey

```

1 PRINT"[CLEAR,RVS] 64 RESTORE/RESET DISABLE - METE KESEN[SPACE2]"
2 IF PEEK(65534)<>72 THEN PRINT"[DOWN]THIS IS ONLY FOR 64 MODE!!"
  :END
3 PRINT"[DOWN]THE RESTORE KEY OR THE RESET BUTTON, IF"
4 PRINT"YOU HAVE ONE, WILL NOW CRASH THE 64!![DOWN]"
5 :
10 FOR J=32768 TO 32776:READ K:POKE J,K:NEXT
20 DATA 167,2,167,2,195,194,205,56,48
30 FOR J= 679 TO 687:READ K:POKE J,K:NEXT
40 DATA 238,33,208,238,32,208,76,167,2

```

Fig. 5-1.



64 Delete Right

If you want to delete text to the left of the cursor, you can easily use the Delete key; but if your target is to the right of the cursor, things aren't so easy. The program in Fig. 5-2 converts the F7 key into a Right Delete key. Run it, then whenever you press F7, text to the right of the cursor will be sucked quickly into oblivion.

```

10 REM 64 DELETE RIGHT - LUCY S.TERRIER
15 FOR J=680 TO 725:READ K:CS=CS+K:NEXT:RESTORE
20 IF CS<>4855 THEN PRINT"DATA ERROR":END
25 FOR J=680 TO 725:READ K:POKE J,K:NEXT
30 PRINT"[DOWN]USE F7 TO DELETE TO RIGHT OF CURSOR":SYS 680:END
35 DATA 120,162,181,160,002,142,020,003
40 DATA 140,021,003,088,096,072,138,072
45 DATA 152,072,165,197,201,003,208,014
50 DATA 169,002,133,198,169,029,141,119
55 DATA 002,169,020,141,120,002,104,168
60 DATA 104,170,104,076,049,234

```

Fig. 5-2.

The Stop-Restore combination disables the feature, but it can be enabled again by executing a SYS 680.

Lucy S. Terrier
Alton, Illinois

**64 F7 Apostrophe Key**

When I'm using Commodore's "Magazine Entry Program," I find it very awkward to press Shift and the 7 key to get the apostrophe before the checksum.

To make program entry a little bit quicker, I wrote the program shown in Fig. 5-3. It's a short machine language program that transforms the F7 function key into an apostrophe key. Happy typing!

Brian Evans
Bay Springs, Mississippi

```

100 REM 64 F7 APOSTROPHE KEY - B. EVANS
110 REM MAKES 64'S F7 KEY PRINT AN APOSTROPHE
120 FOR J=50060 TO J+36:READ K:POKE J,K:NEXT
130 SYS 50060
140 DATA 120,162,153,160,195,142
150 DATA 020,003,140,021,003,088
160 DATA 096,072,138,072,152,072
170 DATA 165,215,201,136,208,005
180 DATA 169,039,032,210,255,104
190 DATA 168,104,170,104,076,049
200 DATA 234,000,000,000,000,000

```

Fig. 5-3.

**128 KEY Hint**

I love to type the programs from this book and the column that inspired it, many of which include DATA statements in a distinctive format. To make typing as easy as possible, I set up a function key to produce the framework of the DATA lines, and I use the AUTO command to generate the line numbers. After using the function key to produce a dummy line, I fill in the numbers for the DATA items. Here's how I set up my function key:

```

KEY7,"DATA 000,000,000,000,000,000,000"
+ CHR$(13)

```

John P. Todd Sr.
Stanley, New Mexico

**128 F-key Keeper**

If you redefine your function keys within a program, it's nice to get the original key assignments back when the program is finished. You

can accomplish this easily with the following lines. Execute the first one before redefining your keys, and execute the second at the end of your program.

```
10 DIMK%(255) : FOR J = 0 TO 255 : K% = PEEK(4096 + J)
   : NEXT 655545 FOR J = 0 TO 255 : POKE 4096 + J,K%(J) : NEXT
```

*R. Harold Droid
Seattle, Washington*



128 Auto LOAD/RUN

The F8 key, or any F-key, can easily be redefined to let you use a Directory listing to automatically LOAD and RUN BASIC programs. In direct mode, type

```
KEY 8,"RUN[TAB3,SPACE3]" + CHR$(13)
```

Using a disk directory listing, place the cursor on the line for the program to be run. When you press the F8 key, the program will load and run automatically.

If you use the word LOAD instead of RUN, the F-key will load the program but won't automatically run it. If you like, you can put the key redefinition into a program rather than doing it in direct mode.

*Fred C. Krause
Pittsburgh, Pennsylvania*



128 F-Key Conversions

Hexadecimal and CHR\$ conversions are stumbling blocks to many who would like to learn assembly and machine language programming, not to mention those who aspire to be BASIC experts.

BASIC 7.0 has four functions (HEX\$, DEC, CHR\$, and ASC) that make these conversions easier, but the complicated syntax of these functions introduces stumbling blocks of its own. Even if you know where the parentheses, quotes and dollar signs are supposed to go, your fingers may be slow or uncooperative in putting them there.

If you run the program shown in Fig. 5-4, however, your conversions will be as close as your F-keys. The program redefines keys to print the complete syntax for four different conversion functions. When the F-key is pressed, the necessary characters are printed on the screen, with the cursor blinking in the position of the first character of the argument.

```

10 REM 128 F-KEY CONVERSIONS - TRACY
20 REM USES F-KEYS TO MAKE HEX/DEC/ASCII CONVERSIONS
30 Q$=CHR$(34)
40 KEY 1,"? DEC("+Q$+"[SPACE4]" +Q$+" ) [LEFT6] "
50 KEY 3,"? HEX$([SPACE5]) [LEFT6] "
60 KEY 5,"? ASC("+Q$+" "+Q$+" ) [LEFT3] "
70 KEY 7,"? CHR$([SPACE3]) [LEFT4] "

```

Fig. 5-4.

Type the argument, hit Return, and bingo, you've made your conversion.

*Simon Tracy
Charlottesville, Virginia*

○☆ Ultimate 128 F-key Disabler

The easiest way to disable the 128's function keys is with POKE 828,183. This redefines the vector to the routine that interprets the function keys. After doing the POKE, all keys revert to their standard character codes, but the function key definitions are not touched. To restore these definitions, use POKE 828,173.

By the way, the standard character codes for the F-keys are:

F1 = 133	F3 = 134	F5 = 135	F7 = 136
F2 = 137	F4 = 138	F6 = 139	F8 = 140
RUN = 131	HELP = 132		

These numbers are returned by the ASC function for keystrokes captured by GET or GETKEY.

The codes for F1-F7 also apply to the Plus/4; on that machine, the HELP key's code is 140.

*Charles Lavin
Coral Gables, Florida*

☆ 128 Caps Lock Reader

To read the position of the 128's Caps Lock key, use

C = PEEK(1) AND 64

If C = 0 then the Caps Lock key is down; if C = 64, then it is up.

*Daihung Ho
East Moline, Illinois*

○☆ **Quick Escape from 128 Graphics**

When your 128 is in one of its graphics modes, you can quickly get back to text mode by pressing the Stop key, then any letter key and Return. The text screen will pop up with a harmless Syntax Error.

*Stan Smith
Gladwin, Michigan*

☆ **Another 128 ESCape Sequence**

There's an undocumented function in the 128's screen editor. It is ESC ESC (pressing the Escape key twice in succession). This sequence disables reverse characters, quote mode, and the insert mode caused by leftover INST keypresses.

*Charles Lavin
Coral Gables, Florida*

Related Tips in Other Chapters

Because the keyboard is used in virtually every computer operation, virtually every tip relates to it in one way or another. The following tips relate particularly closely to the keyboard, to keyboarding, or to the use of individual keys:

Chapter 2	Program Typing Tip Keystroke Descriptions in Listings
Chapter 9	Joysticks and Keyboard Errors
Chapter 12	Another TI\$ SAVER First Program in the Directory Loading from the Directory 128 Easy LOAD and RUN
Chapter 13	Quantum Link Multitasking Quantum Link Repeating Keys
Chapter 23	Automatic Line Listing Sensing Special Keys Detecting Disabled Stop Key
Chapter 25	64 Keybeeper
Chapter 26	64 Binary Key Decoder
Chapter 32	64 Sixpack Plus
Chapter 33	128 Keys in 64 Mode

Video Displays

“Video, vides, videmus. Clarius videbimus.”

Latin Saying

This chapter relates to hardware only—your monitor or your TV set. Once that hardware is producing a quality picture, you can use the programming tips in Chapter 24.



Tips for Better Video

Even if you have a top-quality monitor, your video picture can leave something to be desired. If you use a television set, the situation is likely to be worse. Here's a collection of proven ideas for enhancing the picture on all sorts of video displays.

- If you're looking at text, try turning down the color control on your display. The resulting black and white picture is often easier on the eyes.

- Adjust your monitor controls for non-text applications, too—you'll be surprised how effective they are.

- Try using a different color combination on the screen. You can get a black screen by executing `POKE 53280,0 : POKE 53281,0` on the 64 or 128, or by executing the proper `COLOR` statement on the 128 or Plus/4. Holding down the Control key and simultaneously pressing 2 will give you nice white characters.

- Use a commercial glare screen to reduce glare, or wear polarized sunglasses to achieve a similar effect. (Properly selected, the shades can also make a fashion statement).

- Wavy lines and other interference can often be eliminated by moving the computer's power supply as far from the monitor as possible. If the interference persists, try reversing your power plug in its socket. This is impossible with three-prong plugs or polarized two-prongers, but on the others it sometimes gives impressive results.

- With TV sets, the antenna cable is the most likely entrance point for interference. Use a single piece of shielded coaxial cable between the computer and the TV's antenna terminals. Your local electronics store should be willing to make this cable up for you. If your set has a shielded antenna connector, use it. If it has only screw terminals, install a coax matching transformer. Eliminate all unshielded wiring in the antenna circuit, and eliminate the TV/computer switchbox. Then you'll have to switch from TV to computer by physically switching the cables, but if you have the right cable connectors, that's not so hard to do.

- If you **MUST** retain the TV/computer switchbox, run a wire from the case of the switchbox to a good ground. In houses with three-prong electrical sockets, the screws on the outlet covers are usually very well grounded.

- If you don't mind doing the unusual, try putting aluminum foil or a flat aluminum object under your computer. Fantastic results have sometimes been reported.

- You can use antenna filters to reduce screen interference on TV sets. Radio Shack's 15-581 and 15-582 have been known to give good results. Because you want to remove any interference being sent into your set, connect one end of the filter as close to the TV's antenna terminals as possible. Connect your switchbox and/or computer to the other end of the filter.
- If your computer interferes with other TV sets connected to the same antenna, one of the above filters can also be helpful. In this case, connect it between the common antenna line and your TV/computer switchbox. Unlike the previous tip, the idea here is to filter out whatever your computer may be feeding back into the antenna.
- Finally, if you have interference that you suspect is coming in over the power lines, try a line filter such as Radio Shack's 15-1111.



Shield Your TV Cables

If you use a TV set rather than a monitor for your video display, you might have trouble with wavy lines and the like on your screen. The key to eliminating them lies in the cable that connects the computer to the TV. It should be in good condition, and totally shielded if possible. If your TV is set up for a 300 ohm twin-lead, you should have a 75 to 300 ohm coupler between the terminals and your computer cable. It's often useful to coil the cable and tape it into place. It usually helps if you take the TV/computer switchbox out of the circuit, or use some other method to remove the TV antenna connection whenever you're computing.

Radio Shack and similar stores have a wide array of connectors, cables, switches and filters for TV and video connections. The store manager can help with your selection.

Lucy S. Terrier
Alton, Illinois



Modify Your TV Switchbox

I have found an easy way to minimize the wavy lines often found when your computer is used with a television set.

The secret is to shorten the wires that connect the switchbox to the TV antenna terminals. Usually they are about four inches long, which is much longer than is needed. I cut them down to an inch, then hook

them to the antenna terminals. I've done this every place I have moved, and it never has failed me yet.

Robert Huey
Atlantic City, New Jersey



Monitor Glare Reduction

Most monitors for Commodore computers have shiny glass faces on their CRT's, so there's often plenty of glare to contend with. A nylon-mesh screen filter can reduce this glare to a very acceptable level. The filter's fine black nylon mesh is stretched over a strong but lightweight frame that fits over the CRT bezel on the monitor. The one I use is an Acco 55571, and is available for under \$30 in office supply stores.

To accommodate the huge diversity of monitor screens, these filters come in various sizes and with many different mounting schemes. Mine included over a dozen mounting pieces—Velcro, double-sided foam tape, plastic brackets, etc. The Velcro worked well with my Commodore 1902, even though the monitor face is curved and the filter is flat.

The nylon is tough and can be cleaned with soap and water.

Cecil C. Serfent
Chesapeake Beach, Virginia



Sight Improver

If computing leaves you with tired eyes and fuzzy vision, have your sight checked by an eye doctor. I encountered this trouble myself, but put off having an eye exam because I had always had perfect vision. Finally I had my eyes examined, and sure enough, I needed glasses, but only for reading and computing. My glasses look good, they feel good, and most importantly they keep my eyes from blurring at the computer.

The eye exam itself was painless, inexpensive and interesting, and my insurance paid for most of it. The bulk of the work was done by a computerized auto-refractor, which does in one second what the doctor used to do in thirty minutes.



Screen Marking Hint

The special marking pens made for overhead projector transparencies are perfect for making temporary markings on your CRT screen. Their ink adheres nicely to the glass, and is easily removed with

a damp cloth. The Vis-a-Vis brand, made by Sanford, is available at any large office supply store, in extra fine, fine and broad points.

*Barbara A. Sander
Pittsburgh, Pennsylvania*



Screen Photography

Here are some tips to use when making pictures of your computer screen. I get good results with Ektachrome color slide film (ASA 64 or 100) in a tripod-mounted 35mm single lens reflex camera. I darken the room enough to eliminate reflections on the screen, and I aim for a 1/2 second exposure. Because a slight underexposure produces sharper text, I select an f-stop a half step above the one indicated by my built-in exposure meter, using a manual exposure override if necessary. (He sent along some slides, and they really look good!—LFS).

*Steve Haber
Winnipeg, Manitoba
Canada*



RGB Monitor Extension

The Radio Shack #270-1750 ten-foot joystick extension cable works perfectly as an extension for the 128's RGB monitor.

*David G. Veatch
Warren, Michigan*



1902 Monitor Tips

There are two protruding levers on the sides of this monitor, about 8 inches from the bottom front edge. If you pull them downward, they become feet that raise the back of the monitor with respect to the front. This really cuts down on reflected glare, and is a real eyesaver when your monitor sits on your desk or a raised shelf.

The little jack on the left side by the speaker is for headphones. Use them, and family members will feel better about your computing. If you get a 1/8-inch to 1/4-inch mono adapter at Radio Shack or a similar store, you can use your high-quality stereo headset to listen to your computer. The results are so good they may startle you.

*Tracey S. Sander
Pittsburgh, Pennsylvania*

☆ 128 Video Monitor Switch

If you use a monochrome monitor with your 128, it is capable of displaying both 40- and 80-column screens without internal changes. But when you change column sizes, you still have to switch the monitor cable from one computer output to another. I solved the problem by making up a switchbox with a SPDT switch. The rotor of the switch goes to the monitor cable, while the two other contacts go to the computer's two video output connectors.

The 40-column monochrome output comes from pin 1 of the eight-pin DIN connector, while the 80-column monochrome comes from pin 7 of the DB-9 subminiature connector. On both connectors, pin 2 is ground, and should be connected to the shield of your video cable. Be careful when locating these pins, because the pinout diagrams in the manual are somewhat ambiguous. Most connectors have tiny pin numbers molded into their plastic insulators. Look for them before making your connections, and double check them before soldering.

*Reed Seitz
Ellington, Connecticut*

Related Tips in Other Chapters

These tips will also help you optimize your video display:

Chapter 4	Glare-free Lighting Monitor Shelf
Chapter 32	Household Items in the Computer Room Video Test Pattern Generator
Chapter 24	Programming the Video Display

(In English, the Latin saying at the head of this chapter means "I see, you see, we see. We will see more clearly." It was translated by David Seward of the University of Pittsburgh.)

Printers and Printing

“He who first shortened the labor of copyists by device of movable types was disbanding hired armies, and cashiering most kings and senates, and creating a whole new democratic world: he had invented the art of printing.”

Thomas Carlyle
Sartor Resartus
1833-1834

When you have a printer, you're part of that whole new world; with a printer, you have power.

But with their DIP switches, escape sequences and control codes, it can be a challenge to make printers to do your bidding. This chapter will help you to meet it.



Printer DIP Switches

If you use a non-Commodore printer and interface with your system, both those units are likely to have several tiny switches that must be set properly for everything to work right. They're called DIP switches (after the integrated circuit Dual Inline Package standard that their size and contacts conform to), and you can set them with a pen point or other similar instrument. Because DIP switches are both important and confusing, some words of explanation are in order.

DIP switches configure your equipment for different types of operation by determining things such as number of characters per inch and number of spaces skipped between lines. Usually, the switch settings are read when power is first applied to the printer or interface; if you change them, you must turn things off and back on before the changes take effect. Also, in most cases, the DIP switches merely establish defaults; software commands can change the parameters they control. Such commands are usually given by sending control codes or escape sequences to the printer.

Many printing irregularities are the fault of improper DIP switch settings. These settings are always covered in the instruction manual, but are seldom explained in detail. Usually there is a simple chart with very brief descriptions of each setting; to use your system effectively, you should have perfect understanding of these settings. If you can't understand them, show your manual to a friend who's further along than you.

If both a printer and an interface are involved, you must coordinate the settings of the switches on both units. For example, both will frequently have DIP switches with identical functions, such as adding a linefeed after a carriage return. If you're not careful, you could get two linefeeds instead of one.

Similarly, you must coordinate your DIP switch settings with the default settings of the software you're using with the printer/interface combination. Getting three spaces between lines? Most likely the software, interface and printer are each adding one!



Power for Your Interface

Most printer interfaces come with an ungainly power cord that attaches to your computer's Cassette port. If you're capable of soldering wires to PC boards, there's a good chance you can do away with the cord. Here's how:

On many printers with a Centronics-type interface, pin 18 of the printer connector is wired to the + 5 volt power supply, and can be used

to power your interface. If your printer has this feature, it's not very hard to eliminate the need for the Cassette port connector. Just cut the wire that goes to the Cassette port connector, and resolder it to pin 18 of the printer connector.

You must be very careful about wire colors when making this change, especially with Cardco interfaces. On mine, the power wire is white where it connects to the Cassette port connector, but it's blue inside the interface. You also need to be careful when identifying pin 18, and when soldering to the interface board. None of these things are very difficult, but they do require that you work as cautiously as you can.

*Victor H. Pitre
Castle Shannon, Pennsylvania*



Using Multiple Printers

In many cases, it's possible to connect more than one printer to your Commodore computer. All that's required is that the printers have different device numbers. Although we commonly think of the printer as being device #4, it can have any legal device number. The available numbers depend on the hardware in the printer itself.

The 1520 Printer-Plotter, for example, is configured as device #6. As long as you have an extra serial port available, you can connect the 1520 at the same time as your regular printer.

Commodore's DPS1101 can be set up as any device between #4 and #11, and the MPS801 can become #4 or #5, depending on the setting of a back-panel switch.

To print to a device other than #4, just OPEN a channel to it, then print to that channel. For example:

```
OPEN 5,5 : PRINT#5,"HELLO"
```

will print the word HELLO to whichever printer has device #5. The second 5 in the OPEN statement is the device number of the printer. The first 5 is the file number, which is used as the argument in the PRINT# or CMD statement. You can make the file number anything you'd like to make it, but for the sake of convenience, it is usually chosen to match the device number. To illustrate, here's an alternate way of printing to device #5:

```
OPEN 4,5 : PRINT#4,"GOODBYE"
```

Many printers and disk drives have, as you may have noticed, two serial ports. That's so you can connect many different devices to your

serial bus. If you have the right sort of printers, those many devices can include more than one of them.

*Thomas Wilkins
Franklinton, North Carolina*



Two Printers with One Computer

It's well known that you can attach two printers to your serial bus as long as they have different device numbers. But even if they have identical device numbers, you can still do it, as long as only one of them has power applied at a time.

On Commodore printers, this means just flipping the main power switch off. But on printers that use an interface, it's the interface's power that makes the difference, because it is the interface that the computer sees as the printer. You can control the interface's power either by installing a switch or unplugging its power connection.

*Al Morrison
Baltimore, Maryland*



Printer Pause Control

Most of today's printers have an Online/Offline button. When you push it, it stops the printer from printing or advancing paper. Many times this is easier than attempting a pause from your computer keyboard.

*Cory Moore
Westlock, Alberta
Canada*



Printer Hex Mode

The Star SG-10 and many other printers have a useful hex mode, in which everything the printer receives, including control codes, is printed out as hex digits; the effect is sort of like a Machine Language Monitor.

Hex mode is very useful for debugging bad print routines, because it allows you to see exactly what the printer is receiving—control codes and all.

To activate hex mode on the SG-10, first turn the printer off. Then hold down the FF and LF buttons while reapplying power; keep them

depressed until the beeper sounds. The printer is now in hex mode until it is initialized or turned off.

*Barrett Stridiron
Charlotte Amalie, Virgin Islands*



Extra Print Modes

I use an Okidata 120 NLQ printer, and have recently considered upgrading, partially because of the lack of an italicized font. While investigating upgrade options, I learned that many printers with italic fonts cannot use them in their NLQ modes.

So for most normal printing, I decided to use my existing printer's NLQ mode in place of printing italics. I just set up my program's printer driver by entering the codes for NLQ-on and NLQ-off when the driver prompted me for the italicizing codes. This produces surprisingly nice results, and lets me take more advantage of my printer and word processor.

Another feature often forgotten is 'elite', or 12 character per inch (CPI) printing. Many dot matrix printers produce better-looking output at 12 CPI than at the normal 10. When using elite, remember to adjust your margins—left at 12 and right at 84 will produce a six-inch printed line with one-inch margins.

*James T. Peeples
Texarkana, Arkansas*



Printer Paper Alignment

You can save time and wasted paper by noting and marking various "alignment landmarks" on your printer. The first step is to learn exactly where the printhead operates, and how to line up your paper so the head will print in the desired place on the sheet. Print a line while watching the printhead carefully, observing the relationship between the paper, the printing, and a convenient landmark on the printhead mechanism. Now put a pencil mark on the paper and try aligning it so the printing falls exactly on your mark. After a few tries, you should be very good at lining things up. So you won't forget, make a note and/or diagram and keep it somewhere handy. Sometimes the handiest place is on an adhesive label attached to the printer.

Adhesive labels are also useful for marking margin locations for different sized paper. On my Commodore DPS1101, there's a piece of masking tape with marks for aligning 8 ½-inch letterhead and 9 ½-inch

perforated paper. If I align the paper with the marks, I know that the left margin will always be in the same place when I use the printer for word processing. Because the left margin is always in the same place, so is the center of the printed line; consequently everything I center is really centered well, regardless of which paper I'm using.

I've also made an artificial landmark for the top edge of 11-inch paper. When I align the paper with my landmark, I know that printing will always start the same distance below the top of the sheet. I've set my word processor's top margin and page lengths accordingly, so my top and bottom margins are always even and perfect.

Finally, I have marks on my printer for aligning mailing envelopes. Because envelopes are a different width and length than normal paper, it used to be hard to put the address exactly where I wanted it; but with my landmarks, it's a snap. For perfect placement of a three-line address, I put the envelope's top left corner at my special "envelope landmark", set the left margin at 32, and print.

Most of my landmarks were located by trial and error, then permanently located by making a line on a self-adhesive label. Mechanical parts of the printer can serve as landmarks, if they are stationary and properly located.



Feeding Bottom Feeders

The bottom feed printer may be the greatest thing since sliced bread, because it eliminates the possibility of paper feeding out and back into the printer; but feeding the paper into that bottom slot can be an absolute pain, as I discovered the second time I spent fifteen minutes loading a new stack of paper.

The solution is to tape the first sheet of the new stack to the bottom of the last sheet of the old stack. Just make sure the tape doesn't overlap the feed holes or protrude past the edges of the paper. I've been using this method for several months now, and I've had no trouble loading paper.

*John R. Stevenson
APO New York, New York*



Easier Printer Control

Most printers have a complicated series of escape sequences or special characters that control their special features. I can never remember all of them, so I've made up some lines to define them all

in terms that are easy to remember. My printer is a Star NX-10C, but a similar technique can be used with any other machine.

Here's part of my little program:

```
10 ES$=CHR$(27) : EX$=CHR$(14) : CE$=CHR$(15)
20 UN$=CHR$(45) : AA$=CHR$(48) : BB$=CHR$(49)
```

These variable names have mnemonic value, as can be seen from their definitions:

ES\$ is E`S`cape

EX\$ is for EX`P`anded printing

CE\$ is to C`A`ncel Expanded printing

UN\$ is used for UN`D`erlining

AA\$ clears underlining and several similar features

BB\$ sets the features cleared by AA\$

To use this system, I take the lines defining my printer variables and put them at the start of my main program. Then to go into, let's say, expanded printing, I just have the program PRINT EX\$. Enabling underlining requires an Escape followed by a CHR\$(45) and a CHR\$(49). As long as my variables are defined, I can enable it by PRINT ES\$;UN\$;BB\$. Simple, eh?

Kenneth Burrows
McMinnville, Oregon



MPS803 Tractor Modification

If you have Commodore's MPS803 printer with the optional tractor feed unit installed, you've probably noticed that the tractor feed unit makes the paper advance knob hard to get to. This is due to a small plastic guard on the left end of the tractor feed, which always seems to get in the way when you want to position your paper. Because the guard is held on by one small screw, I removed mine, and now I can get at the paper advance knob. I put the screw back in its original hole, and I saved the plastic guard where I can get it if I ever need it.

Matt Zuhlke
Lake Mills, Wisconsin



MPS1000 in NLQ

Here's how to put this popular printer into NLQ mode without having to set the DIP switches. First, turn on the printer and adjust the paper. Then, in direct mode, enter this:

```
OPEN 4,4,7 : PRINT#4,CHR$(31);CHR$(141); : CLOSE 4
```

Next load your word processor and the document to be printed. The output will be in the NLQ mode.

Many word processors include some method of sending printer setup codes, either from the document or from the word processor itself. If yours is one of them, sending the two CHR\$ codes may be enough to do the trick.

If you'd like, you could include the above line in a printer setup program.

*R. V. Taylor
Little Rock, Arkansas*



Blue Chip Printers

These popular printers use the same ribbons as the Commodore MPS802. If you have a program that doesn't list the Blue Chip as one of the printers it supports, set the program up for the Epson FX.

*John Kreps
Grapevine, Texas*



Okimate Head Cleaning

I tried several different chemicals to clean the printhead on my Okimate 10, and I found the best by far to be a Kodak product for cleaning photographic film.

Kodak Film Cleaner #195-6986 leaves the printhead super clean and residue free. Be sure to heed the warnings on the bottle!

*Perry Wilson
Des Moines, Iowa*

Related Tips in Other Chapters

Chapter 8, "Printer Supplies," contains many tips on paper, ribbons, printwheels and other accessories for your printer. These tips also apply:

Chapter 4	Inexpensive Cleaning Fluid
Chapter 12	Making Directory Printouts
	Double-spaced Directory Printouts
	Improved Directory Printouts
Chapter 22	128 CP/M and Printers
Chapter 30	Sprite Graph Maker
	Label Printer
	Graph Paper Maker
Chapter 33	Commercial Software and Your Printer

Printer Supplies

*“Fire in each eye, and paper in each hand,
they rave, recite and madden round the land.”*

Alexander Pope
Epistle to Dr. Arbuthnot
1734

They must have been replacing a ribbon!

As every printer owner can testify, supplies cause their share of headaches. Additionally, some of them are expensive or maddeningly hard to find in stores around the land. This chapter supplies some relief.

○△☆ **Mark Your Fanfold Paper**

When using fanfold paper, it's very desirable to know where the pages end so that the paper can be properly lined up in the printer.

There's a very easy way to mark the top and bottom of each sheet of paper in a box. Just take a red broad tip permanent marker and mark a stripe down all four corners of the stack of paper, being careful to keep your ink on the folded edge of the strip of sprocket holes outside the perforations.

The alcohol base of the marker allows the ink to soak through the paper, so at the end of each page a little red line can be seen at all the corners. This red line is the end of the page and can easily be used to align the paper in the printer.

When the printing is done and the sprocket strip is torn off at the perforations, the red lines come off with the sprocket strip, leaving the pages themselves unmarked.

This simple procedure takes about two minutes with each box of paper, and has been a great convenience to me for years.

*Bernard J. Sussman
Washington, D.C.*

○△☆ **Home-made Fanfold**

When feeding single sheets into my printer, I used to bypass the out-of-paper switch with another sheet of paper, which was really quite a bother. Now I chain the single sheets together with the new removable type of Scotch tape. It's available in any store, and it peels off without tearing or leaving any marks.

*Don Smith
Wheaton, Maryland*

○△☆ **Paper Saver**

Some stores sell plastic paper carriers that allow unperforated single sheets to be used in printers with tractor feed. Not only do these carriers let you print on standard sheets or letterhead, but they can help save paper as well.

Many printers such as the MPS801 waste a sheet of paper at the start of each job, because the printhead is several inches below the tractor mechanism. To place the printhead at the top of a sheet, you must run the preceding sheet most of the way through the printer; but if you have

one of the plastic paper carriers, you can use it to make up the missing inches.

Steve Hoffman
Address Unknown

○△☆ **Inexpensive Heat-sensitive Paper**

If you use the Okimate 10 or other similar printer, you may have had trouble finding an accessible source of inexpensive paper. My local K-Mart sells Sharp heat-sensitive paper (EA-1LS5) for about \$7.00 for a pack of 100 sheets. It works fine in my Okimate, and of course I don't need to use my ribbon when I'm printing in black only.

Joe Borysowicz
Cicero, Illinois

○△☆ **Better Paper for Okimate Color Printing**

When printing in color on the popular Okimate 10 printer, transparent sheets work much better than smooth paper. They are the same size as regular paper, and are available from most office supply stores. Ask for the transparencies used in overhead projectors.

Once the sheet is in the printer, set the darkness control to the darkest position. When the printing has finished, put a sheet of white paper behind your transparency, and notice the beautiful improvement.

Dave Danielson
Brookfield, Illinois

○△☆ **Labels and Gummed-up Printers**

Those self-adhesive pressure sensitive labels are marvelous for addressing letters, but disastrous if they become stuck to the inside of your printer. You can prevent disaster by paying careful attention to the labels before you feed them through.

You can make sure sticky corners aren't sticking up by laying the label stock face-down on a table and smoothing it out from the back. If the backing paper is torn or bent, throw that section away and use some other labels.

Finally, never try to back labels out of your printer. To remove a string of labels, cut the string before it enters the back of the printer,

then feed the unused labels forward. Believe me, it's much better to waste a few labels than to get one stuck in your printer.

*Kevin Healey
Lansdale, Pennsylvania*

○△☆ **Cheap Printer Labels**

Have a printer? Have a label-printing program, but have no stick-on labels? If you don't need too many labels, you can print them on regular paper, cut them out and use a glue stick to attach them wherever they are to go.

If you're not up to the space-age adhesive technology embodied in the glue stick, old-fashioned Elmer's Glue will work almost as well.

*Shane D. Dolha
Regina, Saskatchewan
Canada*

○△☆ **Colorful Printouts**

I have one of those nice programs that makes borders, designs and greeting cards on my printer. By using colored paper, I can get a very nice effect in many cases. I can add to the effect by using colored markers to highlight or color in certain areas of the designs.

I made my biggest step forward, however, when I bought several colored ribbons for my printer. Now I put in a colored ribbon and print my border. Then I change ribbons, reload the same piece of paper, and print another part of the design. I repeat this for as many colors as I have ribbons, and the result is a full color picture from a non-color printer and software.

There is one caution here—most computer stores don't carry colored ribbons. I found mine at a local store that specializes in ribbons for various types of office equipment; some mail order houses also handle them.

*Arthur McMahon
Escondido, California*

○△☆ **Home-made Envelopes**

You can make your own envelopes for the greeting cards created by "The Print Shop" and similar programs. Take an envelope that fits the cards you print, carefully break its seams and trace it on cardboard.

Cut out the template and, using a blank sheet of paper, trace around the template. Cut out the envelope and fold it to size. Two swipes of a glue stick produce a perfect envelope for your cards, and more glue stick seals it up for mailing. No more hunting for hard-to-find funny-sized envelopes.

*Patricia J. Kercel
Henderson, Nevada*

○△☆ **Greeting Card Envelopes**

If you use the popular “Print Shop” or “Print Master” programs to make up greeting cards, you might have trouble finding the right size envelopes for them. I had that problem until I learned what to ask for.

Go to a well-stocked stationery store and ask for #5½ Baronial envelopes. Many stores carry these in white and ivory, and they are sometimes available in other colors as well.

*John Swingle
Seattle, Washington*

○△☆ **Keep Two Ribbons**

Because printer ribbons quickly lose their like-new blackness, I always keep two of them on hand. One stays in the printer, and I keep it going until it’s dry. The other, a like-new one, stays in my drawer until I need it for a high-quality printing job. I use it for that job only, then return it to the drawer until it’s needed again. When the “everyday” ribbon finally wears out, I replace it with the moderately-worn ribbon from my drawer, and replace that ribbon with a brand-new one.

*Luke A. Babey
Port Washington, New York*

○△☆ **MPS801 Ribbons**

The Commodore MPS801 and the Comrex Model 220 printers use identical ribbons. If you own one of these and can’t find ribbons, look for a ribbon for the other.

*Vinicio Perez Coto
San Jose, Costa Rica*

○△☆ **MPS803 Ribbons**

If you can't find these ribbons locally, look for a Brother M-1009 ribbon, which seems to be an exact replacement.

Gary Greer
Sylacauga, Alabama

○△☆ **Daisy Wheel Ribbons Explained**

Commodore's DPS1101, like many other letter-quality printers, uses a standard IBM Selectric II ribbon cartridge, available at most office supply stores. There are many varieties of Selectric II ribbon on the market, with prices that vary widely; not all of them are suitable for your printer. The store clerk can help you choose the right one, probably after consulting her catalog. In the meantime, here's an overview of the most important features:

The least costly ribbon has a pink knob and pink leader. With each strike of the daisy wheel it advances one full character. Because the ribbon advances so fast, the cartridge has a rather short life. This ribbon's print quality is extremely sharp—sharper, in fact, than that on most office typewriters.

A better ribbon for most computerists is the more costly variety with a blue knob and blue leader. This ribbon advances only a fraction of a character with each strike, giving a five to ten times greater ribbon life. (It's not five to ten times more expensive). The print quality, while not as sharp as on the pink-knobbed ribbon, is as good as that on most office typewriters. I prefer it for most purposes because it looks a lot more "normal".

The most common ribbons are the so-called "correctable" varieties, with orange or yellow knobs and leaders. You probably don't want to buy these because your printer can't take advantage of their correctability, and they aren't designed for long life.

○△☆ **Emergency Ribbon Replacement**

If your last printer ribbon gives up the ghost and can't be immediately replaced, you can be back in business in seconds. Remove the paper from your printer and replace it with two sheets separated by a sheet of carbon paper. Nothing will print on the top sheet, of course,

but if you've set your impression correctly, the carbon copy should be perfectly legible.

*Dale Blake
Whiteman AFB, Missouri*

○△☆ **New Life for Old Ribbons**

When your cloth printer ribbon can no longer make clear, dark characters, don't throw it away. You can still use it to print five or ten pages by putting your printer into near letter quality (NLQ) mode, if your printer has such a mode. In NLQ mode, each character is struck twice with a slight offset. With a good ribbon, this gives you exceptionally black characters that lack most of the "dot matrix look". With an almost-dead ribbon, it gives you legible printing instead of something too light to see.

*Chuen K.
McPherson, Kansas*

○△☆ **Rejuvenating Cloth Ribbons**

When the ribbon gets kinky or fuzzy, but there's still lots of ink left, you can use a low-tech electrical device to rejuvenate it. Take the ribbon out of the cartridge, smooth it down and place it between two sheets of paper. Set an electric clothes iron on a low setting and iron the ribbon back to usefulness.

*Steven Sellore
Saint John, New Brunswick
Canada*

○△☆ **Ribbon Re-inking**

It is messy and tedious to reink my MPS801's inker cartridge one drop at a time. I avoid the mess by putting a few drops of ink into a 35mm film container or a pill bottle, then dropping my inker into it, felt tip first.

*Ben Johnson
Address Unknown*

○△☆ **1520 Plotter Supplies**

If you have Commodore's very nice 1520 Printer/Plotter, you know that finding paper and pens can be quite frustrating; but the cure is as

close as your nearest Radio Shack store. Radio Shack's PC-2 printer takes the same supplies as the 1520, and these supplies are available at any Radio Shack.

Black pens are #26-1480A, and assorted colored pens are #26-1481A. Each is about \$3 for a three-pack. The paper is #26-1428, and comes in a three-roll box for about \$5. If your local store doesn't stock these items, ask the manager to order them for you.

*Michael Hale
Gaithersburg, Maryland*



DPS1101 Printwheels

I treasure my Commodore letter-quality printer, but I've had problems getting printwheels to fit it. It's easy to find printwheel suppliers, but it's hard to know which of their many types will work in the DPS1101. Although most office supply stores have printwheels in their catalog, they seldom keep them in stock, so you can't compare them with your original. Even worse, the suppliers' cross-reference tables almost never include Commodore, and the DPS1101 manual doesn't identify suppliers.

After months of searching, I've finally found what to get. GP Technologies, a very large printwheel manufacturer, has a series of wheels that's perfect for the DPS1101. Many office supply stores carry their printwheels and other products. If you call GP at (800) 523-1809 or (201) 722-7165, they can direct you to your nearest dealer.

The product you're looking for is called Dual Plastic Printwheel for Royal/Adler (Group 02). There are seven different typefaces available, in 10 and 12 pitch plus proportional spacing. Price is in the \$25 range. I know these wheels work with the DPS1101, because I've tried one of them in my own. My typeface is Orator, a 10-pitch face with very large letters. A few of its special symbols (ten, to be exact) are different from those on my original wheel, but most of them are ones I've never used. The single exception is the up-arrow, which appears on the original wheel as a caret or circumflex. On the Orator wheel, for some reason it's a triangle.

Other manufacturers surely have wheels for the DPS1101, but I haven't run across them. GP says they are for use on most Royal and Adler typewriters and Juki printers. That information might let you cross-reference to another maker's wheels, but be careful about the most.

If you want to put custom characters on your printwheel, that can be done for \$35 per printwheel plus \$15 per character. For details, telephone Dramco at (800) 223-0979.



More on DPS1101 Printwheels

I've bought and tested these wheels for my Commodore DPS1101 daisywheel printer:

Montgomery Wards typewriter department carries JUKI monoplasic printwheels priced at \$16.99 each. Their Mini Majestic 12/15 is the smallest type size I've found.

Venture's typewriter department carries Royal/Triumph Adler monoplasic printwheels for Royal Alpha 2001, Adler Satellite II and Olympia Portable Electronic typewriters. For another \$16.99, their Elite-Modern 10/12 pretty well rounds out my printing needs.

*Walter J. Upham
Kansas City, Missouri*

Related Tips in Other Chapters

These tips also relate to the subject of Printer Supplies:

Chapter 3	Free Paper, Free Disks, Free Equipment!
Chapter 4	Easy Printer Stand Nice Marker Printer Ruler Printer Scrap to Your Notebook Printer Scrap to the Microwave
Chapter 7	Printer Paper Alignment Feeding Bottom Feeders Blue Chip Printers Okimate Head Cleaning
Chapter 11	Better Disk Labels

Of Mice and Joysticks

“When the master is away, the mice play on the table.”

Romanian Proverb

When the master is at home, they *still* play around up there, with their long-tailed cousins the joysticks. These tips will improve their games.



1350 Mouse as Joystick

The new Commodore 1350 Mouse not only works great with programs designed for it, but it works as a joystick with Doodle and other joystick-oriented programs. My advice is to try the mouse where fine, slow action is called for, but stick with your joystick for the hard-core action games.

*Reed Huish
Mesa, Arizona*



1351 Mouse in Joystick Mode

It isn't mentioned in the user's guide, but you can select the joystick mode after powering up the computer. Just hold down the right button as you plug the mouse into the port.

This is very useful if you have loaded your program but installed the mouse in the wrong port or didn't install it at all. It saves you having to repower the computer and reboot the program.

*Juan Rodriguez
Abington, Pennsylvania*



Label Your Joysticks

It may be obvious, but lots of people don't do it. If you keep two joysticks plugged into your computer, label them so you can tell which joystick is plugged into which port.

I use a piece of tape on the joystick, marked with the number 1 or 2. Others have used bands of tape on the joystick cables. Try narrow masking tape or adhesive tape, putting one band on Joystick 1 and two bands on Joystick 2. If you put identical bands on the cables near the plugs, you'll eliminate confusion when you connect and disconnect the joysticks.

*Tony Brantner
Munhall, Pennsylvania*



Use the Proper Joystick

Having problems getting high scores in games like Track and Field and others that require rapid left and right joystick movement? Maybe the trouble is with your joystick.

Though you'll never see it in the specs, some brands of joysticks have a much smaller "turning radius" than others. The ones with the small radius are the best for rapid back and forth movement. The Atari joysticks, unsophisticated as they are, are in this desirable class.

*Shane D. Dolha
Regina, Saskatchewan
Canada*

○△☆ **Troubleshooting Joysticks**

If your joystick seems to be bad, it may be that something else is really the problem. Here are several steps you can take to narrow it down to the joystick.

First, be sure the joystick is in the proper port. Some programs require it to be in Port 1, while others look for it in Port 2. Second, try another joystick. If it doesn't work, then the first joystick is probably OK. In this case, you may have blown the joystick chip; and third, try the joystick with another program—maybe your software is at fault, and not the joystick.

Once you know your joystick is really bad, you might try taking it apart. Most joysticks are simple mechanisms, and if you're careful you can disassemble them quite easily. Once you can see inside, look closely for loose wires or broken connections. If you find them, you can often fix them, even without using tools.

*John Kuhne
Newmarket, Ontario
Canada*

○△☆ **Joysticks and Keyboard Errors**

If you leave your joystick plugged in when you're not running joystick software, you may get an unpleasant surprise. Depending on the port it's in, moving the joystick or pressing the button can make the computer think that you've pressed one of the keys on the keyboard.

To prevent this happening at unanticipated moments, unplug your joystick whenever it's not being used.

*Stewart Terrier
Alton, Illinois*



Using Non-Commodore Joysticks

Many brands of joysticks will work well for your Commodore computer. The ones from the old Atari 2600, which nearly everybody in the world owned at one time, work perfectly. Coleco is another company whose once widely-sold joysticks are compatible with your machine. You can often get used joysticks for pennies at flea markets and garage sales.

*The Terrier Twins
Alton, Illinois*



Joysticks for Southpaws

If you're left-handed, you know how hard it is to press the firebutton on standard joysticks. By making some wiring changes, you can put the button in a better position for your hand.

Disassemble the joystick and see where the cable is connected to the printed circuit board. There will be six wires going to push-on connectors that are easily removed. Take note of their order on the board, which should be brown, white, black, blue, green and orange.

Pull off the connectors, then reinsert them in this order: blue, brown, black, green, white, orange. (Notice that the orange and black wires don't move).

Reassemble your joystick, hold it with the firebutton to your right, and attack your favorite game.

*Mary Lee Resnick
Butler, Pennsylvania*

Related Tips in Other Chapters

These tips also involve joysticks and their operation:

Chapter 14	Joystick Port Connectors
Chapter 16	Joystick Pieces
Chapter 33	Using an Extra Hand

Datassette Recording

*“Anon they move
In perfect phalanx, to the Dorian mood
Of flutes and soft recorders.”*

John Milton
Paradise Lost
1667

The earliest Commodore PETs had a built-in software recorder: a cassette deck. In subsequent models it moved, anon, to an external box: the Datassette.

Today, when other manufacturers have eliminated cassette tape capability, the 64 and 128 remain fully compatible with the Datassette. It's a good storage device for those who can't afford a disk drive, and a good backup device for those who can.

This chapter's single tip is a distillation of ten years' experience in using Commodore tape recorders.



Datassette Tips

The Datassette is Commodore's special cassette tape recorder. If you don't yet have a disk drive, it's your only offline storage device. You can use these hints even if your Datassette is collecting dust at the bottom of your closet.

If it is collecting dust, you should consider bringing it out into the light. On two occasions I made errors that caused my word processor to hang up the serial bus, which prevented me from saving my valuable document. I hooked up my Datassette and saved my work to tape, then reset the computer to get out of the hang-up. I also make tape backups of my most important programs, so if (when?) my disk drive is out of service, I can still enjoy my computer.

For those who use the Datassette on a regular basis, Radio Shack and other stores have a wealth of useful products for handling cassette tapes. A bulk eraser is very useful, as is a kit to repair cassettes. There's no end to the variety of boxes and storage cabinets, either. I prefer the soft plastic boxes to the hard ones because they are more durable and less expensive, and can easily be used for sending tapes through the mail.

When buying tapes, it's good to buy the shortest ones available, because it takes so long to search for programs on a long tape. The C-10 tapes, with five minutes of recording time on each side, are wonderful if you can find them. C-30's are eminently satisfactory, and widely available at low cost. Inexpensive brands such as Radio Shack's Concertape are perfectly satisfactory for Commodore use. Avoid C-120's like the plague, because the ultra-thin tape they contain is very subject to tangling.

To write-protect a tape, punch out the little plastic tab on the left side of the back of the cassette. To overcome the write protection, put a piece of tape over the punched-out hole. Scotch tape is fine, and the write-protect tabs used with floppy disks are perfect.

Before using a tape for the first time, it's a good idea to fast forward it to the end, then rewind it. This process, which you might want to do twice, balances the tension and avoids potential tape jams. It's also a good idea to write down the tape counter reading at the end of the tape; it will be useful when you need to estimate how much tape is left on a fairly full cassette.

When using a tape, it is NOT necessary to run it past the clear plastic leader before recording. Commodore's excellent recording scheme takes the leader into consideration, and records only after it has passed the head.

To avoid Load Errors, it's a good idea to avoid saving programs with identical names. Otherwise your computer, when finding the name of an old, partially-erased program, can think it's found the program you want. Other techniques for avoiding these maddening errors are to bulk erase old tapes before using them, and to degauss your tape head after every few hours of use. Head demagnetizers are available at Radio Shack or any electronics store.

On the 64 and 128, you can hear your tapes as they load by executing a POKE 54296,15. Turn the volume on your monitor all the way up, and you'll hear some interesting things. With practice, you can tell from its sound whether a given tape is good or bad.

And finally, when your computer says it has FOUND a program, you can quickly start the tape rolling again by pressing the Spacebar or the Control or Commodore keys.

*Jerald Brown
Erlanger, Kentucky*

Related Tips in Other Chapters

This tip shows another use for your Datassette:

Chapter 33 **Bypassing Dongles and Protection Keys**

Floppy Disks

*“The ever-whirling wheel
Of Change; the which all mortal things doth sway.”*

Edmund Spenser
The Faerie Queen
1590

These tips are for the floppy disk itself — how to buy it, format it, label it, store it and keep track of it in your library.



Disk Buyer's Guide

If you're confused about what sort of disks to buy, this tip will help you out. It discusses floppy disks in general, and makes specific recommendations on what to purchase for your Commodore.

While there are many specifications for disks, the most important ones refer to number of sides, density, and type of sectoring.

Many disk drives, such as the Commodore 1541, have only one read-write head, which writes on the bottom side of the disk. Others, like the 1571, have two heads—one on the bottom and one on the top. Diskette manufacturers sell single-sided disks and double-sided disks, one for each type of drive. The manufacturing process is identical for both, but single-sided disks are only tested on one side, while double-sided disks are tested on both. Because of the extra testing, double-sided disks are a little more expensive, but the price difference is lessening all the time. At this writing, it's less than 10% at many stores.

When floppies first came out, they had a certain capacity for information storage, which matched the requirements of the equipment of the day. Then the equipment improved, and the new drives could record twice as much information per square inch of disk surface—if the disk was capable of storing it. To accommodate the new drives, disk manufacturers introduced higher quality disks, which they called *double-density* to differentiate them from the older product. Because of their higher quality, they could be used on either the old or the new drives. Later on, special *quadruple-density* or *high-density* disks began to be made; in general, they can only be used on drives of a special design.

Sectoring is either hard or soft, and refers to the number of little holes punched in the rotating part of the disk, near the center hub. Soft-sectored disks have only one hole, while hard-sectored disks have several. You can use either kind in your Commodore drive, because it doesn't use the little holes at all. Hard-sectored disks are rarely found in today's marketplace.

Today, the double-sided double-density (DSDD) disk is by far the most common in the marketplace, because it's the standard disk for IBM PC's, XT's and their clones. High-density (HD) disks are found in stores that cater to IBM AT users, and single-sided single-density (SSSD) disks are getting harder and harder to find.

SSSD disks will work fine with the 1541 and with the 1571 in 1541 mode (its normal mode when used with a 64 or with a 128 running as a 64). In its double-sided mode (normal with a 128), the 1571 should

be used only with DSDD disks. High-density disks may not work at all in your Commodore.

Because of their wide availability at reasonable prices, I recommend buying DSDD disks for all Commodore drives, even though SSSD or SSDD disks work fine in the 1541.

There are many brands of disks, and some can be more than twice as expensive as others. I have used all kinds at great length, and have never had a problem with even the cheapest disk; the poorest floppy disk is a very high-quality item, and perfectly satisfactory for Commodore computing. Nevertheless, I stick to name-brand disks for my own purchases, because I don't want to take any chances. Which route you take is, of course, completely up to you.



How to Make "Flippy" Disks

If you have a 1541 or other drive that only uses one side of the disk, here's an old trick for using the other side. All you do is to cut another write-protect notch just opposite the old one, in the area where the manufacturer's label is usually placed. When it is properly placed, the new notch will be a mirror image of the old one.

Using another disk for a template, make a penciled outline of the notch, then cut it out with a razor blade or delicate pair of scissors. Be careful not to go outside the lines, or you might ruin the disk by cutting into the rotating part.

Once your new notch is complete, you can insert the disk upside down (oval read-slot first, manufacturer's label downward) and format it like a brand-new disk. You now have the equivalent of two disks, with two directories, one on one side and one on the other.

There are some dangers inherent in this trick, and you need to be aware of them. The flipped disk rotates in the "wrong" direction with respect to its jacket, and it was never designed to do so. Neither was it meant to have its flip side written on, especially in the reverse direction. For these and other reasons, there's a chance you'll lose some data from the second side of the disk. Nevertheless, many computerists accept the risk, and most of them report perfect results from all their "flippy" disks. Because of the risk, some people use flippies only as backups, or for programs whose loss would not be serious.

You should be very careful when using a double-notched disk in a 1571 or other double-sided drive. These drives write on both surfaces of the disk, no matter which side is up, so it's easy to make a mistake that overwrites the "other" side.

*Kathleen Mead
Westerville, Ohio*

○△☆ Flippy Disk Identifier

Do you use disks that have been converted to double-sided “flippies” by the addition of a second write enable notch? Have you noticed that when such a disk is in the drive, you can’t look at it and tell which side is being used?

I solve the problem by putting a silver write-protect tab on the top edge of the disk, about an inch from the left side of the disk. When the disk is in the drive, I can see the silver tab. If it’s on the right, the normal side is being read. If it’s on the left, it’s the “flippy” side.

*Joseph J. Jernigan
Chattanooga, Tennessee*

○△☆ Assigning Disk ID’s

When you format new disks, you may not put much effort into avoiding duplicate ID codes. Under most circumstances, that’s perfectly all right.

It’s still wise, however, to implement a consistent policy that avoids duplication of disk ID’s, because there are two circumstances where duplicate ID’s can cause trouble:

- 1) Some software that uses multiple disks does not run properly if those disks have identical ID’s.
- 2) Many disk library organizing programs are keyed to the disk ID and cannot catalog more than one disk with the same ID.

A good way to avoid duplicate ID’s is to make up a card the size of a diskette and store it in the back of your disk file. Each time you format a disk, write the ID on this card in its proper alphabetical place. If you follow some pattern like AA, AB, AC, AD and so forth, you’ll never duplicate an ID again.

*Arleigh
Quantum Link*

○△☆ Disk Library Tips

Most experienced computerists have systems and procedures for keeping track of their disks. Here are some of the better ones we’ve seen over the years.

- As soon as you buy a new box of disks, open it. As soon as you open a box of disks, format every disk in the box. As soon as you format a disk, attach a blank label to it. This process will ensure that you never have to stop what you're doing to format a disk.

- If you can avoid it, never give two diskettes the same ID, because some programs can become confused by duplicates. A good system is to assign ID's in a regular alphabetical sequence: AA, AB, AC, AD, and so on. Wherever you keep your disks, if you also keep a card that shows the ID's you've assigned to date, you'll easily be able to avoid duplicates.

- Reserving certain disks for certain categories of programs often makes it easier to find what you're looking for. Common categories are utilities, sound and music, games, word processing files, and so forth.

- Put your most-frequently-used program first on the disk, where it can be loaded by `LOAD"*",8` or a similar automated command.

- Don't try to use all the space on a disk, because you need space to update programs and data. Leaving 50-100 blocks free is wonderful insurance against overflow.

- Keep one or two disks on hand exclusively for use in program development. As you work on new programs, save all your intermediate versions to one of these disks. Then, when a program is "finalized," put it on a different disk and erase all the previous versions. This saves wear and tear on your good disks, and lessens the chance that you'll ruin one of them by mistake.

- Print out each disk's directory, and keep the hard copy close to the disk itself. Some people tape it to the disk envelope, while others keep it inside the envelope.

*Mary Lee McCafferty
Butler, Pennsylvania*



Disk Tabs and Labels

When you buy a box of disks, it usually contains a modest supply of labels and write protect tabs. Here are some suggestions for using them and replacing them when they're gone.

Some people code their disks by the position of their labels. For example, games disks could have the labels on the right, word processing disks in the center, and utilities disks on the left. If you combine this

technique with color coding via the labels, you can have a very sophisticated identification system.

Speaking of color coding, many computerists use colored electrical tape or the colored dots from stationery stores for the purpose. Your imagination is the only limit, so there's hardly any restriction.

You can also use electrical tape to make write-protect tabs. A single roll makes dozens or hundreds of tabs.

When you run out of disk labels, you can buy colorful preprinted replacements at Radio Shack and elsewhere.

Also check your office supply store's selection of general purpose labels. These are available in many sizes and colors, and for a few dollars you can purchase a long-term supply. I've had good success with the sheets of mailing labels designed for copy machines—they are the right size, they adhere well to the disks, and one box holds over 2000 labels.

*Lucy S. Terrier
Alton, Illinois*



Better Disk Labels

There's nothing magic about the preprinted labels that come with your floppy diskettes. When they're gone, consider buying some self-adhesive labels from the stocks at an office supply store. These stores have many different sizes and colors, and for a few dollars you can buy a year's supply.

Some of the labels come in boxes, while others come on rolls or in computer-perforated fanfold. While the standard 1-inch x 3 1/2-inch address labels are wider than normal disk labels, they're a plentiful and economical substitute; and if you're a stickler for neatness, you can use your printer to write on them.

*Pearl Pickard Hertzler
San Diego, California*



Home-made Disk Envelopes

If you've lost a disk envelope and are looking for a serviceable replacement, you can make one up yourself. Cut a blank piece of typing paper in half from left to right. Fold up 3/4-inch of one end of the resulting piece, and tape both sides of the 3/4-inch overlap. It isn't much, but it works!

*Dave Egts
Johnstown, Pennsylvania*



Colorful Disk Envelopes

Try this little trick when you use one of those programs that prints out a pattern for making your own disk envelopes:

Load and run your favorite graphics utility such as Print Shop or Print Master. Create a design with nothing but a graphic, covering the entire sheet of paper. Then put the paper back into the printer, with the graphic facing the printhead.

Load your disk envelope pattern-maker, and print the envelope pattern over the graphic. When you cut out your stylish new disk envelope, you'll see a look of envy on everybody's face.

*Earl Harrison
Address Unknown*



Disk Mailers

It's not safe to mail a disk unless it has some protection from being folded by an aggressive postal worker. Commercial mailers are serviceable but expensive, and they aren't always strong enough to withstand determined folding.

A very rugged mailer can be made from two squares of corrugated cardboard, slightly larger than a disk. Line them up so their corrugations are at right angles to each other, then put the disk between them like the meat in a sandwich. Tape on all four sides completes a package that is almost impossible to bend, and that's safe in an ordinary envelope.

Another strong one can be fashioned from Styrofoam. Tape your disk to an inch-thick piece, put it in an envelope, then dare the mailman to bend it.

*Al Sperling
Mount Lebanon, Pennsylvania*



Thick Diskettes

Is it hard for you to get 50 diskettes into a disk holder that claims to be able to store that number? The problem is most likely with the disk envelopes, rather than with the disks themselves.

If your disk envelopes are made of thin cardboard, rather than of paper or indestructible Tyvek, the disk and its envelope will be as much as half again as thick as a disk in a thinner envelope. (My Verbatim Datalife's used to come in Tyvek, but the newest box came in cardboard. The new ones don't fit well in my storage boxes - LFS).

If your storage problem is extreme, you can buy Tyvek envelopes and throw the cardboard ones away.

*John Kuhne
Newmarket, Ontario
Canada*

○△☆ **Creative Disk Storage**

While there are many special boxes for storing and transporting disks, common containers will often do the job for much less money. Take a few disks to the kitchen department of your discount store, and you're sure to discover some inexpensive alternatives—some with airtight seals.

Another useful kitchen/disk product is the Zip Lock sandwich bag, which perfectly holds one or more disks while protecting them from dust and fingerprints.

Wherever you keep your disks, it's useful to keep a supply of labels and write-protect tabs as well. I cut the tops from two 6 x 9 inch mailing envelopes, making smaller envelopes six inches wide and about five inches high. They are about the size of disks, and they fit nicely in the back of my disk file, where one holds labels and the other holds write-protect tabs.

*Margaret Ittel
Los Angeles, California*

○△☆ **Disk Storage**

An old 45 RPM record holder of the steel wire type makes a good diskette holder and desktop storage device.

*Peter Natale
Golden's Bridge, New York*

○△☆ **Temporary Disk Storage**

When using programs that require frequent swapping of disks, you never know where to put the disk that's waiting to go into the drive.

I stand it up in the space between the horizontal rows of keys on my computer.

*Brandon McWhorter
Arlington, Texas*



64C and 128 Temporary Disk Storage

The slots on the rear of the top surface of these computers are convenient places for temporary storage of diskettes. Just stand the disk vertically in the slot; the slot edges will hold it there in a nearly vertical position. Because there are nine slots, each of which can hold three disks, your computer can hold up to 27 disks in this way.

*Tyler A. Rosewood
Regina, Saskatchewan
Canada*

Related Tips in Other Chapters

Chapter 12, “Disk Operations”, has tricks for using the floppy disks you’ve bought, formatted and kept track of here. Chapter 31, “Disk Utility Programs”, contains stand-alone programs for performing disk-related wizardry.

This tip is also related to the topic:

Chapter 4 Household Items in the Computer Room

Disk Operations

*“My joy, my grief, my hope, my love
Did all within this circle move!”*

Edmund Waller
On a Girdle
1664

Your disk drive is vital to every computer activity. This chapter is chock-full of tricks for naming files, working with directories, loading programs, and using the drive in other interesting ways.



Temporary Program Names

When I'm developing a program, I save my work every fifteen minutes or so, to ensure that I always have copies of the latest version and everything that led up to it. While this is a valuable and prudent method of operation, it has a few drawbacks: it tends to clutter the disk with programs that are very similar to one another; I have to think up a unique name for each version; and when the program has reached its ultimate state of development, I have to erase all those preliminary versions.

A very simple program-naming procedure makes it easy to manage all these programs. I just name my first version TRY1, my second TRY2, and so on. When I make a major revision, say right after TRY5, I name it TRY10. The next major revision, maybe after TRY14, is TRY20. It's very easy to remember the last number I saved, so it's just as easy to know the name for the next one. If I ever lose track, a quick look at the directory shows exactly where things stand, because the latest version has the highest number.

When it's time to erase these old backup versions, I can get rid of entire blocks of them at once. Scratching TRY? deletes everything from TRY1 through TRY9. Scratching TRY2? does the same for everything numbered in the twenties. The only thing to be careful of here is that I don't erase my final version! To be on the safe side, before scratching anything I always save the current version with some name other than TRY.

Another way to name different versions is to use the last line number you have completed in the program. When you save the version that ends with line 370, call it "-370". When you save the one that ends with line 510, call it "-510", and so on. Once the program has been completed, save it under a "real" name, and use S0:-* to scratch all programs whose name starts with a dash.



Program Naming System

My system of naming programs uses the wildcard capability to cut down on the amount of typing. When I get a disk, I name the files so each is prefixed by a unique two-digit number: 00, 01, 02, and so on. Then when I want to use file 03, I just:

```
LOAD "03*",8
```

If I am unable to remember the number, I enter:

```
LOAD "??filename",8
```


where filename is the name of the program I'm looking for, less its unknown prefix.

This system takes a little effort to set up, but the convenience it provides is worth it.

Steve Moon
Duluth, Minnesota



Date and Time in Filenames

Because programs are so often changed and improved, it's very useful to know the date and time that a given version was created. It's easy to accomplish that goal by including the date and time in the program name. Because a program name can have up to 16 characters, if you use four characters for the date and four for the time, you'll have up to eight characters left for the actual name of the program.

The system I use is to type the program name (up to eight letters) followed by an eight-character date-time group. The name WARGAMES02141100 would indicate that this version of WARGAMES was created on February 14 (0214) at 11:00 A.M. (1100). A version created on February 15 at 1:32 P.M. would be named WARGAMES02151332 (note the use of 24-hour military time format).

In four years, I've never been unable to describe a program in the eight characters allowed by this system. The benefits of having the date and time have revealed themselves countless times, as I've encountered different versions of the same program on the same or different disks, and have always been able to know, without doubt, just which is the most recent version.

Lucy S. Terrier
Alton, Illinois



TI\$ in Filenames

When developing programs, it's wise to save your work periodically, to be sure it won't be destroyed by a power failure or other disastrous event. If you use a new disk for this purpose, the revisions will appear sequentially in the directory, letting you keep better track of your progress. I generally save all my revisions, at least until the program is finalized, because I never know when I might need something I threw away in one of the early revisions.

By using the variable TI\$ to name each revision, keeping track of them becomes even easier. When you type SAVE TI\$,8 then press Return, your program will automatically get a unique six-character name.

(All right, just probably unique. You could have saved another version exactly 24 hours ago). This method doesn't work with the 128 or Plus/4's DSAVE command or with the Left Arrow SAVE command used in the 64's DOS Wedge program, but the automatic naming feature is more than worth the extra keystrokes used to type the SAVE, the comma and the drive number.

With DSAVE, you can add the time to the end of another filename by using DSAVE "filename" + TI\$.

You can achieve even more by setting TI\$ equal to the correct time when you turn on your computer. Each program saved using TI\$ will then have a name corresponding to the time that it was saved. If it's 11:55, you set TI\$ from direct mode by entering TI\$ = "115500". The "00" is for the seconds, and they must be part of the string. Remember, too that using your reset button sets TI\$ = "000000".

I've recently adopted this technique in my own work, and it has taken the tedium and confusion out of saving my backups. If my development work takes more than one day, I use a new disk each day, labeling it with the appropriate date. When it's time to save the finished version of my program, I save it on another disk, using a meaningful plain-English name. When I no longer care to save the previous revisions, I just reformat their disks, eliminating the ID code from the formatting command. In seconds, the disk is ready to be reused.

*Cecil C. Serfent
Chesapeake Beach, Virginia*



Another TI\$ SAVER

You can use the above technique on your 128 or Plus/4 by redefining a function key as follows:

KEY 7, "DSAVE" + CHR\$(34) + "16FEB" + CHR\$(34) + " + TI\$" + CHR\$(13)

The "16FEB", of course, is changed to the date I'm doing the saving. This method gives me the date as well as the time in my program names, and makes it easy to scratch one day's work by using SCRATCH "16FEB*".

*Jim Loomis
Van Etten, New York*



First Program in the Directory

I often like to use a menu program that lists the disk directory and lets the user easily choose his program. Although this menu program

is the last one saved to the disk, it's best to have it as the first one in the directory, where it can be loaded with the asterisk wildcard or by using the 128 or Plus/4's RUN key.

With a little advance planning, it's easy to put the menu program in the first position on the directory. When I format a new disk, I immediately write a dummy program consisting of just a line number and a colon. I save it with a name like PLACEHOLDER, and of course it goes into the first position in the directory. When I'm ready (much later) to save my menu program, I scratch PLACEHOLDER and save the menu program. Its directory entry takes the place of that for PLACEHOLDER—first in line in the directory.

Marvin1
Quantum Link



Loading from the Directory

Here's an old trick that newcomers will be happy to learn. If your directory is on screen, you can use it to make program loading easier. Just move your cursor to the desired program's line on the listing, and type LOAD in the first four spaces of the line. (You'll be writing over the digits for the number of blocks in the file.) Then use the cursor-right key to move past the quote just after the filename, and type a comma, the number eight, and a colon. Press Return, and your program will load automatically!

Like so many useful tricks, this one is easier to do than to describe. Try it once and you'll see how easy it is; and yes, if it's a machine language program you want to load, you can add a comma and the number one between the eight and the colon.

Maged Abdel Wadood
Cairo, Egypt



Easy-load Filenames

To help my children with loading and running, I name my programs using a filename of ten characters or fewer, followed by these six characters:

[SHFT SPACE]
[CMDR D]
[CMDR U]
[CMDR D]
[CMDR E]
[SHFT @]

When the directory is listed, it will show the ten-character filename, followed by ",8,1:". To load and run the program, the kids just move the cursor to the proper directory entry, then press Shifted Run/Stop.

*Dennis E. Smith
Houston, Texas*

☆ **128 Easy LOAD and RUN**

It's easy to load and execute BASIC 7.0 programs by entering RUN "programname" from direct mode; but here's an even easier way.

Save your program with a name that ends in a shifted space and a shifted @ sign. Such programs can be loaded from their directory listings by moving the cursor to the listing and pressing F6.

If the program has been named already, you can use the RENAME command to change its name to one ending in shifted space and shifted @.

*Frank Klein
South Elgin, Illinois*

○△☆ **Disk Directories by Filetype**

When loading a directory, most people know how to use the asterisk and question mark to restrict the load to a certain group of filenames. You can also select the directory for a single filetype by adding an equals sign and the first letter of the filetype after the pattern. For example, to load the directory of all SEQUential files, type:

```
LOAD "$0:* = S",8
```

To load the directory of all SEQUential files starting with the letters ABC, type:

```
LOAD "$0:ABC* = S",8
```

(You can get by without the 0, but it's a good habit to always include it.)

The filetype principle also works with the 128's DIRECTORY command, and will usually work with whatever other directory loading system you may have.

On the 64, but not on the 128, you can use the filetype designator to load a SEQ or USR file into memory, just as though it were a program. To put the SEQ file KATHY into the BASIC program area, just execute:

```
LOAD"KATHY,S",8
```

Notice that the S for the SEQ filetype is inside the quotes, appended to the name of the file.

*Jerry Jordak
Middlefield, Ohio*



Making Directory Printouts

To print a disk directory, put the desired disk in drive 8, then enter these lines:

```
LOAD"$",8  
OPEN 4,4 : CMD 4 : LIST
```

Wait for the printer to finish, then:

```
PRINT#4 : CLOSE 4
```

Remember, the only legal abbreviation for PRINT# is P Shift-R.



Double-spaced Directory Printouts

By using a file number greater than 127, you can make your printer put a blank line between each item in the directory. The syntax is:

```
LOAD"$",8  
OPEN 128,4 : CMD 128 : LIST
```

When the printing has stopped, execute:

```
PRINT#128 : CLOSE 128
```

*Lucy S. Terrier
Alton, Illinois*



Improved Directory Printouts

When you list your directory to the printer, the disk name prints in reverse field (white letters on a black background), which is not too legible on Commodore printers, and which totally confuses some others.

The reverse field effect can be eliminated by a simple direct mode command, executed after the directory has been loaded. Here are the appropriate commands for the various Commodore computers:

64	POKE 2053,32
128	GRAPHIC CLR : POKE 7173,32
PET/CBM	POKE 1029,32

PLUS/4 and C16	GRAPHIC CLR : POKE 4101,32
VIC-20 (Unexpanded)	POKE 4101,32
VIC-20 (Expanded)	POKE 4613,32

The POKE is to an address five bytes above the start of the BASIC program area, and replaces the reverse field character with a space. On some machines, the previous execution of certain graphics statements causes the start of BASIC to move, so we have included a GRAPHIC CLR to undo the move. In most cases it can be omitted without any adverse effect.

*John Gidusko
Fern Park, Florida*



Reading Protected Directories

On some protected disks for the 64, the directory is protected from being listed. You can often overcome this protection by putting the computer into 128 mode and using the DIRECTORY command. Sometime you even get a message in addition to the directory—my “X-15 Alpha Mission” game displays a message about hackers when the above process is carried out.

*Dwaine Shank
Damascus, Maryland*



Backup Disks and Heartbreaks

I’ve been working with home computers since 1979, and I’ve owned a dozen different models. Over that period, I’ve written or edited about 200 magazine articles, plus 2,000 tips like the one you’re reading now. Most people would say I know what I’m doing.

One day I was reviewing, organizing and upgrading the 125 programs that had appeared in my *Commodore Magazine* column to date, with an eye toward collecting them all in this book. As I finished with each program, I SAVED it onto a master disk I was using to accumulate the collection. Being an experienced and cautious computerist, every half hour or so I made a backup copy of that disk. Being interested in speed and efficiency, I kept both the master and the backup in front of me on my desk, each carefully identified on its label.

Because I’d been working on this project all day, both my main disks were very full of programs. As part of my work, I was also using several other disks, each of which I would reformat from time to time.

Imagine my surprise when, in a momentary lapse of concentration, I reformatted my master disk!

“Well,” I thought, “no harm done, since I’ve wisely made a backup.” Imagine my further surprise when I discovered that, in a previous lapse of concentration, I had also reformatted my backup!

There I sat, experienced, knowledgeable and cautious, just having erased every bit of work I’d done since getting up that morning. Eight hours of meticulous program development, done on almost a hundred different programs, absolutely wiped out!

Instead of crying (which I felt for a while like doing), or cursing (which I was very tempted to try), I started to reconstruct my work. It was easier the second time through, but it still took over six hours.

Like most computer tragedies, mine would have been avoided if I’d used just a little more caution. I did the right thing in making a backup, but I stopped a little too soon. Every time I made that backup, I should have taken it far from my computer, where foolish hands couldn’t easily get to it. In fact, I keep a disk storage box in the top of my linen closet, exactly for this purpose. That box holds the original of every commercial disk I own, plus the originals of all my best creations. I keep it far from my computer so I won’t be victimized by momentary lapses of attention.

Last weekend, in fact, that box held all but one of the important backups in my life; and you see what happened to that one!

The moral of the story is simple and very important. You should repeat it aloud at least twice:

- 1) MAKE A BACKUP OF EVERY IMPORTANT DISK YOU OWN.
- 2) KEEP THE ORIGINAL IN A SAFE PLACE OUTSIDE YOUR COMPUTER ROOM



Easy Disk Drive Reset

When your drive gets temporarily confused, you can reset it without touching the keyboard or risking the loss of your program. Just reach behind the drive and turn its power switch off. Wait a few seconds, then turn it on again. Your drive will be reset, and your program will be perfectly safe, because the computer itself isn’t affected.



Head Stick Fix

If your disk drive should get jammed at or above track 35, making your drive useless, there’s a very easy way to correct the problem. Just

take that strange piece of cardboard that came inside the drive (you do still have it, right?) and put it back in the slot. The tab on the back of the cardboard will push the head back to track 1, and your drive will be ready to use!

*Perry Rothermel
Waltham, Massachusetts*



Disk Head Unsticker

Sometimes when working with copy-protected disks that use tracks above track 35, the disk head gets stuck up there and all further attempts to use the disk drive fail, even after turning the disk drive off and on. The easiest way to return the drive to normal is to initialize it by entering:

OPEN 15,8,15,"10" : CLOSE 15

The disk head will knock one time, then everything should work properly.

*Ed Haymore
Sunnyvale, California*



Transporting Disk Drives

If you are shipping or transporting your disk drive, you should always insert the cardboard head protector that came with it; but if you don't have the cardboard protector (and who does?), just use a diskette, inserted into the drive sideways. This not only protects the drive head, but also provides a safe and convenient place to store a disk.

*Brian Slack
Menlo Park, California*



Avoiding Loading Slowdown

I use a Tymac CONNECTION interface with my 128 system, and it's one of my favorite pieces of equipment. One day two friends pointed out that it took my drives two minutes to load a program that their systems handled in twenty seconds! Directories in my word processor also took forever to load.

At first, I thought the problem was with my drives or in the computer itself, or maybe even a bad cable; but after swapping components with my friend's system, I discovered that the culprit was my interface. When I disconnected it, everything loaded quickly and accurately.

The problem was in the CONNECTION's SRQ line, which is used by the fast serial bus as a bi-directional fast clockline. The slow serial bus doesn't use it at all. To correct the problem, I removed the cover from the end of the interface connector and clipped the black wire from pin 1.

I'm not sure, but maybe this problem occurs with other interfaces too. If your system is very slow in the 128 mode, this could be the fix you need.

*Marshall Freedland
Miami, Florida*



64 DOS Wedge Slows Programs

The Wedge is very convenient for making disk access easier, but its use extracts a time penalty. Because the computer is constantly checking for wedge commands, it can't run other programs as fast as if it were disabled. A few quick experiments revealed a slowdown of between 6 and 17%.

To disable the Wedge, enter @Q. To enable it again, enter SYS 52224.

*Stanley L. Anderson
Tishomingo, Oklahoma*



128 DOS Shell Copies

The "DOS Shell" program is the most useful utility that comes on your 1571 Test/Demo disk. Making a working copy is a lot more difficult than it might be, because there's some autoboot code on Track 1, Sector 0 that can't be copied in the usual way.

One way to make a copy is to use the DOS Shell's copy a DISK command to copy the 1571 Test/Demo disk onto a blank disk. It will copy in double-sided format.

Now reformat the disk you just made, being sure NOT to specify a new ID code. The drive will spin for a while, and when it stops, use the DOS Shell's File Copy option to copy DOS Shell from the original disk to your new one.

*Frank Klein
South Elgin, Illinois*



128 DOS Shell Speedup

To speed the loading of the DOS Shell program, insert the disk after you get the READY prompt, then type BOOT and press the Return key.

The program will load in about ten seconds, as opposed to 25 seconds when it loads by resetting with the disk in the drive.

*Jeff McKee
Rochester, New York*

☆ **128 and Locked Files**

Some software uses locked files to protect important information from being erased. (You can tell a locked file by the presence of a “greater than” symbol next to its file type in the directory.)

The 128, when using the 1571 drive, cannot load such files. To load them successfully, you must first unlock them. There are many simple programs to do the unlocking, or you can do it yourself with a disk editor. Go into the directory entry for the file, and find the filetype byte. Change its first nybble from \$C to \$8, and the file will be unlocked.

*W. Anthony Marsh
Somerville, Massachusetts*

○△ **1571/64 Speed Report**

In most applications, the 64 can't take advantage of the 1571 disk drive's speed advantage over the older 1541. Programs, for example, will load into your 64 just as fast from a 1541 as they do from a 1571; but there are areas where the 1571 has a speed advantage over the 1541, even when used with a 64.

While the 1541 takes about 1:12 to format a disk, the 1571 does the job in only 0:43. And there is a similar advantage when validating a disk. One of my disks could be validated on the 1571 in 1:22, compared to 1:37 for the 1541. Unlike formatting time, validating time depends on what's recorded on the disk in the first place.

*Lil Willard
Marietta, Ohio*

○△☆ **1571 Storage Limitation**

The 1571 drive can store twice as much data on its double-sided disks as the 1541. All is not perfect, though. The 1571's directory is the same size as the 1541's—144 files maximum.

Granted, this is a somewhat trivial limitation, unless you're storing many small files on the disk. On the 1541, you could store 144 four-block files and end up wasting 88 blocks because the directory is full.

On a 1571, you could still only store 144 four-block files, but you'd be wasting 752 blocks of the disk — the original 88 blocks plus the entire second side!

*Charles Lavin
Coral Gables, Florida*



Double-sided 1571 in 64 Mode

If you reset the computer in 64 mode, the 1571 disk drive automatically sets itself to single-sided mode. If you want to work in 64 mode and retain the double-sided capacity of the 1571, you can do what I did.

I put the following program as the first one on any disk that's destined for 64 mode programs:

```
100 DIRECTORY : PRINT "PRESS Y FOR 64 MODE" : GETKEY A$
110 IF A$ = "Y" THEN GO64 : ELSE NEW
```

I saved the program as "GO64", then used my autoboot creator to run it automatically when the system boots.

Now when I turn on the computer, a single keystroke gets me going with my drive active in double-sided mode and my computer in 64 mode.

*Larry Hardy
APO, New York*



1571 Quirks

The 1571 disk drive is a double-sided drive, and is capable of formatting disks with 1328 blocks free. This compares favorably with the 1541 drive, of course, which can put only 644 blocks on one side of the disk.

But the 1571 often defaults to a single-sided mode, giving only 644 free blocks on a newly-formatted disk. Worse yet, this mode only recognizes 644 blocks, even if the diskette has been formatted with 1328! Here's the story on controlling the single- or double-sided mode of your 1571:

If you use 128 mode to format a disk, it will be formatted on both sides, with 1328 free blocks. Also, 128 mode will read any 1541/1571 disk, regardless of how it is formatted.

If you use a 128 in 64 mode, the 1571's status depends on how you entered 64 mode. If you entered it with a GO64, the 1571 will behave exactly as it does in 128 mode—it will treat the disk as double-sided, and will also read single-sided disks.

But if you entered 64 mode by holding down the Commodore key as you powered up or reset, the 1571 will behave as though it were a single-sided drive like the 1541. Disks will be formatted with 644 blocks free, and the drive will recognize only the first 644 blocks on any 1328 block disk you may use in it.

The same 644-block limitation exists if you use the 1571 with a 64 or other serial-bus-compatible computer such as the VIC or Plus/4.

But take heart! One simple command will put your 1571 into double-sided mode, regardless of the computer to which it's connected. When you power up in anything other than 128 mode, just execute the following line:

```
OPEN 15,8,15,"U0>M1" : CLOSE 15
```

Your disk drive will now behave in all respects as a double-sided drive. To ensure that this will always be the case with your 128, turn the command into an autobooting program and put it on all your diskettes. You accomplish this, of course, by using the "Autoboot Maker" program found on the 1571 Test/Demo disk.

*Itty Greer
Glenshaw, Pennsylvania*



1571 Motor Shutoff

A bothersome feature of the 1571 is that its motor continues to run for about ten seconds after accessing the disk. You can use this subroutine to turn the motor off in about two seconds. It works with any computer:

```
500 PRINT#15,"M-W";CHR$(53);CHR$(0);CHR$(1);CHR$(1)
510 RETURN
```

Of course, you'll have to execute an OPEN 15,8,15 somewhere before using the subroutine. Then GOSUB 500 after every disk read or write, and the motor will quickly stop.

You can also assign the command to a function key on the 128, if you include the OPEN and CLOSE commands in the key assignment. The assignment statement is straightforward enough, but contains enough punctuation marks to surely confuse the typesetter. I'll leave its exact format up to you.

The routine works by changing the value in the drive's RAM location 53, which is used as a countdown timer for turning off the drive motor.

The normal value is 6, but this program changes it to 1. Don't try changing it to zero, or the motor will not turn off.

*Ray Roberts
Gainesville, Florida*

Related Tips in Other Chapters

Chapter 11, "Floppy Disks", has many tips on the floppy disk itself. Chapter 31, "Disk Utility Programs", contains stand-alone programs for working with disks and disk drives.

These tips also involve disk operations:

Chapter 4	Inexpensive Cleaning Fluid
Chapter 23	Editing SEQ Files
	Efficient Numeric Files

Telecommunications

“Mr. Watson, come here, I want you.”

Alexander Graham Bell
The first words transmitted by telephone
March 10, 1876

Bulletin boards, downloadable software libraries and person-to-person communication linkups have all done their part in making telecommunications the fastest growing area of computing.

This chapter gives tips on connecting your computer to your telephone line, and on getting the most from online services such as Quantum Link.

○△☆ **Modem Connections**

Modems can often be used in ways you won't find in the manuals. For example, in most cases you don't need to plug both the telephone and the line into the back of the modem. It's sufficient to connect the phone line only, and to treat the modem as just another extension on the line. When I discovered that trick, I immediately got rid of the telephone, which gave me extra desk space and eliminated an unnecessary wire.

*Gloria Greer
Glenshaw, Pennsylvania*

○△☆ **Modem Speed Dialing**

Owners of pulse dialing modems are acutely aware of a major drawback—speed! When the computer you're dialing is momentarily busy, redialing those seven or more digits can seem to take forever. If your local telephone company offers the feature called "speed dialing," you can be rid of the delay forever.

The speed dialing feature lets you set up nine or more telephone numbers that you can dial with two digits from any phone on your line. Put your commonly-dialed computer numbers onto speed dialing, and your modem can get by dialing two numbers rather than seven or more.

*Craig Edward Given
Chattanooga, Tennessee*

○△☆ **Telephone Extensions**

If you are tired of moving your computer to the nearest phone jack just to use your modem, you should try installing a semi-permanent extension cord. Parts for such a cord are readily available at Radio Shack and elsewhere, and making it up is easy. Just get the proper wire, tools and connectors, then make up your cord. It's helpful to use rubber coated staples (Radio Shack 64-2876) to fasten the wire to the baseboard.

*Tim Weiss
Address Unknown*

○△☆ **Marking Telephone Cables**

If you use a modem, you probably do a lot of plugging and unplugging of modular telephone cords. There's one from the line to

the modem, one from the modem to the telephone, and often several more. Making the modular connections is easy, but identifying the proper cord can be a real nightmare. They all have identical connectors, and they all are made of similarly colored cable.

I've solved the problem by using a Sharpie brand marker to label the ends of every modular cable in my computer room. The Sharpie is indelible and has a fine point, while the flat modular cable is very easy to write on. If I label both sides of each end of the cable, I can tell at a glance exactly what it is.

*Alvina Gorse
Charleston, South Carolina*



Telecommunications Tips

Try these steps to eliminate line noise that interferes with modem communications:

1) Check your phone wires and other connections. Bad connections can ruin your ability to communicate. Be sure all phones are properly hung up. Shut off your computer and reseal your modem in the User Port. Remove joysticks, speed loaders and other cartridges — disabling them is not enough. If you don't have 3-prong electrical outlets, use an adapter that screws onto the outlet cover's holding screw.

2) Eliminate radio interference. Your phone wire can act as an antenna and pick up stray signals from nearby radio transmitters. The shorter the wire between your modem and the wall, the less chance of picking up radio interference. AT&T Phone Stores sell a Radio Interference Filter (Part #104236484) for under \$12.

3) Get rid of crosstalk. If you ever hear other conversations in the background when using voice, such crosstalk may be fouling up your modem. Try redialing. Try calling later at night when fewer people are using the phone lines.

*Bruce A. Roberts
Bronx Users Group*



1670 Autoanswer Disabler

Commodore's 1670 modem has the sometimes undesirable habit of answering incoming telephone calls to humans. If you leave your modem connected while working on non-modem tasks, you may have encountered the problem.


```

10 REM 1670 AUTOANSWER DISABLE - JONES
20 OPEN 5,2,3,CHR$(8)+CHR$(0)
30 FOR X=1 TO 7
40 PRINT#5,MID$("[SHFT A,SHFT T,SHFT S]0=0",X,1);
50 NEXT
60 CLOSE 5

```

Fig. 13-1.

The program in Fig. 13-1 disables the autoanswer feature. If you put it at the beginning of other programs, they'll be safe from this annoying trouble.

*David M. Jones
Los Angeles, California*



1670 Silencer

As most owners of Commodore's 1670 modem know, the 1670 is an autoanswer modem. This means that when the computer is on and the modem is connected, even if no modem program is running, the 1670 will pick up the line and start to send a carrier. That can be most disconcerting to callers who aren't expecting a computer to answer.

When this particular nuisance strikes, you don't need to turn off the computer or disconnect the modem. Just pick up the telephone and whistle into the mike. If you vary your pitch as if you're calling your dog, the modem will quickly shut up.

*Jason King
West Palm Beach, Florida*



Disabling Modem Autoanswer

Autoanswer modems like Commodore's 1670 will answer incoming calls whenever the computer is on, whether it's telecommunicating or not. Most human callers get very disoriented when a computer answers their calls.

Some computerists avoid the problem by connecting their modems only when actually telecommunicating, or by using software to disable the autoanswering feature every time they power up or reset. Forgetfulness easily defeats either of these techniques, and neither one is compatible with hassle-free computing.

At \$7.95, Radio Shack's 43-127 Ringer Silencer is the perfect solution to the problem—it prevents the ringing signal from reaching the modem, but doesn't disturb anything else.

I keep my 1670 permanently plugged into my 128 and permanently connected, through the switched off Ringer Silencer, to a telephone jack on the wall. Because the modem is always plugged into the line, I can telecommunicate at any time without making additional connections. Because the computer never answers the phone, my human callers retain their pleasant dispositions.

*Allen W. Greer
Glenshaw, Pennsylvania*



Modem Hang Up

With most terminal programs on most systems there is an easy way to force your Commodore 1670 or Hayes compatible modem to hang up the phone and disconnect you immediately. Just type three plus signs and wait for about a second. The modem will send you a zero or an "OK". (Don't confuse it with anything you might be normally reading on your screen.) Once you receive that signal, just type ATH and press the Return Key. The modem will immediately hang up the phone! Those lucky enough to have a Commodore 1670 don't even have to type the ATH.

*Kermit Woodall
Glen Allen, Virginia*



Telecommunications Privacy

Never, never, never use the same password on any two telecommunications systems. Fly-by-night schemes have been occasionally set up to trap the unwary computerist into revealing his secret passwords. By using different passwords for every online activity, you'll be safe from these nocturnal aviators. You'll also lessen the chance of revealing your password to innocent but opportunistic strangers.

*Mike Dryja
Washington, Michigan*



Online Recording

If you subscribe to an online information service, you can use your VCR to save time and money. Just connect the VCR to one of your computer's outputs, then make a tape of your online activities. If something good moves by too fast, you won't have to wait for it to be retransmitted, because you'll have it saved on tape.

It's easy to connect the VCR to your monitor, if you take the time to understand things. You may also need some cables and adapters from Radio Shack or another electronics store. With this explanation, plus the manuals from your computer and VCR, finding the right connections should be a short and sweet project.

Your computer has two "video" outputs: the TV connection from the single-pin RCA connector, and the (two) Video Out connections from the eight-pin DIN connector. The TV connection couples to the antenna of a TV set tuned to channel 3 or 4, while the Video Out connections mate with the Video In connections of a tuner-less video monitor.

Most VCR's have Video In connectors as well as antenna terminals, so you can use either one with your computer. The TV connection is probably the easiest, but may not have the clearest picture. The Video connection may be clearer, but will only be in black and white. (Commodore has two video output wires, luma and chroma. VCR's have only one video input, which should be connected to the luma output.)

You can use either output for your VCR, regardless of which one is in use at present. To connect two devices to one computer output connector, just use Y-connectors to put them in parallel.

*Alicia I. Birch
Clarksville, Tennessee*



Minimizing Telecommunications Charges

If you use one of the online information services, your long distance telephone charges are generally included in the fee you pay for the service. But your local phone charges are another matter. Depending on your own exchange and that of the local access number, your local calls can be free or they can cost you plenty.

Often there are two or more local numbers available for you to call. If you check with your telephone company business office, they can tell you which one will be most economical for you. And if you spend a lot of time online, it may even pay you to change to a different category of local service.

*Moe Demm
Vienna, Virginia*



Online Assistance

If you subscribe to one of the telecommunications services, don't overlook their customer representatives as a source of assistance with all aspects of their service. They are as close as a 1-800 phone call, and

if you can state your problem clearly, the chances are good that they can help you. Before placing your call, research your problem as thoroughly as possible, and make sure you understand exactly what help you need. Write down your problem and your questions, then dial away!

*Charles Brown
Basking Ridge, New Jersey*



Online Reading Practice

I use the online encyclopedia as a source of reading practice for my seven-year-old. The on-screen letters are large, the material is fascinating, and I can choose a subject that he's interested in. When he's doing this reading, not only is he enjoying himself, but he's also learning about reading, computers, and whatever encyclopedia subject he's working with.

*Bobbye Haupt
Newark, Delaware*



Meet Me on Quantum Link

Quantum Link is a low-cost easy-to-use online service, exclusively for Commodore owners. At this writing, it has over 50,000 active members.

Using your modem and free Q-Link software, you can download from a library of 15,000 programs, play challenging online games, and chat with other Commodore owners. You can even communicate online with Commodore's Customer Service department. For further information on Quantum Link, call (800) 392-8200.

I'm very active on the Q-Link, and am always happy to make contact with interested readers. My user name for this purpose is LFS. I'd like to hear from you, especially if you have comments on this book, or if you want to make an online submission to "Tips & Tricks."



Quantum Link Timesaver

Q-Link often leads you through several screens to get you where you want to go. When downloading a program, for example, the author's notes and instructions come up when you choose a file to download. If these screens are long and of no interest to you, you can waste a lot of time waiting for them to finish.

But usually you can avoid the wait. Press F5 as soon as the unwanted screen begins to print, then follow up by pressing F7. The appropriate menu will probably appear on screen, and you can be on your merry way.

The trick doesn't work under every condition, but it's always worth a try.

Anne H. Sander
Boynton Beach, Florida



Quantum Link Quick Messages

Here's a shortcut for posting very brief messages on the Quantum Link message boards. When you're prompted to "enter the subject of the message you would like to send", instead of entering a subject, enter your message. Then press Return, and without entering anything else, press F7 to get the menu. Press F1 to send your message to Q-Link, and see how quickly it gets sent.

Louis F. Sander, Sr.
Boynton Beach, Florida



Quantum Link File Append

During an online session, I often make several disk saves of incoming information. Instead of choosing a unique filename for each of these saves, I give the first one a very simple name, such as FILE. For subsequent saves on the same session, I use the filename FILE,A. The comma and the A cause the new information to be appended to the end of the previously created file, and I can append as many times as I want to, all with this one simple filename!

When I later want to review the information I have saved, I only need to remember one filename—FILE. I can use my word processor to edit it or to break it down into different documents.

If you wanted to get fancy, you could use a more descriptive filename, perhaps including the date. FILE3/15/87 or QLFILE3/15/87, though more difficult to remember, are more descriptive when seen in a directory.

Ranjan Bose
Winnipeg, Manitoba
Canada



Quantum Link Multitasking

While typing an online message or E-mail, you can still interact with others in People Connection or in a conference.

If you press F7 while working on your message, the system will give you a Help prompt for E-Mailing or OLMing, but in fact you'll be back

1) Get all the necessary hardware first. It's a good idea to go with lots of disk storage; at least one megabyte is recommended. Many SYSOPS I know use SFD1001 drives, which give one megabyte of storage on a single 5 ¼" floppy. Hard drives, though expensive, will assure you plenty of storage. A 1200 baud modem is almost an essential, as is a dedicated telephone line with a phone whose ringer can be shut off.

A second computer is good to have, too, in case you get the urge to play California Games when your 64 is running the bulletin board.

2) Pick the right software. The right software package is one that is advanced enough to make its use easy, with plenty of features for the expert user. The popular BBSs in my area seem to like a package called C-Net. It uses lots of memory, but it has the features you need.

3) Keep the BBS operational during its prescribed hours. When you advertise your BBS, always include its hours. It is not good for your reputation when callers get angry voices at 3:00 AM when they expected to get your BBS.

*Sean Machado Leadhead
Address Unknown*

Related Tips in Other Chapters

Here's another tip that might improve your telecommunication operation:

Chapter 16 **Beating the Serial Bus Blues**

Wiring and Connectors

*“The uses of wire are multifarious
and diverse beyond all enumeration.”*

The Encyclopedia Britannica
1894

Is there a rat’s nest of wire in the dust behind your computer? Would you like to know more about the multifarious diversity of jacks, plugs and connectors on the side and back panels of your equipment? Then read the tips in this chapter.

○△☆ **Label that Cable**

If your computer room is like mine, you have between five and ten power cables, all terminating in multi-outletstrips below your computer desk. When the time comes to disconnect something, it's a dusty and frustrating job to find the proper plug.

I've made my life much easier by labeling every power cord in my system. I wound a piece of wide masking tape around each power cord, about two inches from the plug. With that tape as a label, I used a medium-point marker to identify the piece of equipment that's powered by the cord. For removable cords, I repeated the operation on the end that plugs into the equipment.

Now when I need to unplug my 1541, it's easy to find the plug.

Bernie D. Wurm
Perryville, Pennsylvania

○△☆ **DIN Connectors**

The round connectors used for video and serial bus connections on Commodore computers are called DIN connectors, after the German organization responsible for their design. (*Deutsches Institut für Normung* is German for "German Institute for Standards." It's commonly referred to by its initials).

If you buy connectors and wire, it's not difficult to make up cables or extension cords for the various DIN connections in your system. I've made serial bus extenders up to ten feet long, without ill effects from the extra length.

I cannibalize the wire for these from Radio Shack's #270-1705 Joystick Extension Cord, which includes ten feet of very flexible nine-conductor cable. Some of the connectors are also available from Radio Shack, but others need to be bought from an audio or electronic supply store.

When making up DIN cords, you must be very careful about the numbering of the pins. Pin numbers are usually molded into the connectors themselves, but the digits are very small. The numbering system is often not what you'd expect, and even published diagrams have been known to contain errors. The moral of all this is to be very careful, and to rely on the numbers molded into the connectors.

Another confusing point is that published pinouts sometimes don't say which end of which sex connector they are illustrating. Commodore

manuals show the solder terminal end of the male connector, but others may vary. Again, the moral is to be very careful.

Steve Macedo
Timonium, Maryland

△☆ **Joystick Port Connectors**

It's sometimes useful to connect things other than joysticks to these ports. Radio Shack makes a Joystick Extension Cord (#270-1705) which is very useful at these times. You can use it as is for an extension cord, or you can cut it apart and connect your non-joystick device to it directly.

The ports themselves take a connector widely used in the electronics industry, called the DB-9 Subminiature "D". Radio Shack has a variety of these connectors in various configurations, complete with hoods and other cable protectors. Electronics parts stores have even larger varieties.

If you buy one of these connectors, be sure to check their fit in your computer. There's never a problem with the connector itself, but sometimes, especially on a 128 in Port 1, the shell or hood will interfere with the computer's cabinet or with the other joystick port.

If you're stuck with a connector that has this problem, use a joystick extension cord between the computer and the connector.

Mike Swords
Kansas City, Missouri

☆ **128 Inaccessible Joystick Ports**

If you have a software protection "dongle" or other item that won't fit into the 128's joystick port, try using a joystick extension cable. You'll find the details in Chapter 33's "Using Key-protected Software on the 128".

○△☆ **User Port Connectors**

The connectors that mate with your computer's user port are called printed circuit board edge connectors. Edge connectors are made in such a huge variety of grades and configurations that finding the right one can be a real challenge. The one for the Commodore user port is not a very common configuration, which can make your hunt even more difficult.

You can make a serviceable user port connector by taking a hacksaw to a Radio Shack #276-1551. Cut off a 13-pin section from one end of this 22-pin connector, then use pliers to pull out the 13th pin. (That

pin's former space now becomes the end stop for the cut-down 12-pin connector.) Using a similar technique you can cut six pins off the other end, making a nice cassette port connector.

If you'd rather buy an exact-fit connector in the first place, you'll need to find a well-stocked electronics parts store. Ask them for a 12-position dual sided connector with .156-inch pin spacing, suitable for a 1/16-inch PC board. If they stock this size, they may have it in a variety of types and grades. Just pick the one that fits your taste and pocketbook.

Andy Carlson
Youngstown, Ohio



Connector Polarization

Wherever a PC board edge connector is used, the possibility exists to insert it upside down, and thereby to do horrible damage to your system. If you look closely at your user port and cassette port, you'll see that Commodore has cut slots in the PC board between certain pins. These are called polarizing slots, and their purpose is to prevent upside down insertion of the edge connector. To use them, get polarizing pins for your connectors and insert them between the pins in the appropriate places.

The polarizing pins are tiny pieces of plastic or fiberglass that fit in slots between the electrical pins on the connector. When the polarizing pins match the polarizing slots, the connector can be inserted without difficulty. But if you try to insert it upside down, the pins and the slots won't match, which keeps you from making the connection. If you don't have polarizing pins for your connector, you can often make them yourself from thin plastic.

Lucy S. Terrier
Alton, Illinois



Expansion Port Connectors

Connectors for the Commodore expansion port (the one used for cartridges) are not very easy to find in stores. Also, when you find them, they tend to be expensive. So whenever I need such a connector, I cannibalize it from a defunct game cartridge. Not only are these widely available, but the cartridge case makes a nice chassis for whatever gadget I want to connect to the port.

If you do buy your own connector, be very careful about pin numbering. Commodore's system of pin numbers is different from that

used in industry. (Pin 1 in the standard system is pin 22 on the expansion port, and so forth.)

*Biff Sander
Pittsburgh, Pennsylvania*

Related Tips in Other Chapters

If you're interested in wiring and connectors, you might want to look up these tips:

Chapter 4	Connector Labels
Chapter 6	Tips for Better Video Shield Your TV Cables Modify Your TV Switchbox RGB Monitor Extension 128 Video Monitor Switch
Chapter 9	Label Your Joysticks
Chapter 13	Modem Connections Telephone Extensions Disabling Modem Autoanswer
Chapter 17	SX 64 and Printer Interfaces

Troubleshooting

*“Double, double toil and trouble;
Fire burn and cauldron bubble.”*

William Shakespeare
Macbeth
1605-1606

It's a testament to modern engineering that, while most of us deal regularly with operator errors and software glitches, we rarely see a situation where the equipment itself fails to work. When that uncommon trouble bubbles up, the tips in this chapter might help.

○△☆ **Hang-up Recovery**

Recently I had a program that wouldn't run on my 64. When I tried it on a friend's computer and it worked perfectly, I thought I'd have to send my own machine to the repair shop. On investigating further, I found that my voice synthesizer, connected but not enabled, was causing the program to hang up. As soon as I unplugged it, the program worked O.K.

When I told the story to my computer friends, many of them reported similar situations involving printers, interfaces and the like. So my advice is this—when you can't get a program to work, or when certain peripherals seem to hang up, try removing all unnecessary peripherals from your system.

*Steve Ward
Tacoma, Washington*

☆ **128 Apparent Failure**

After coming home from school one day, I switched on my 128 to find it apparently dead. There was nothing on the screen at all. I turned it off and back on again. Still nothing. I checked 64 mode, and everything was normal. After a little more searching to find what was wrong, I noticed that the 40/80 Display key was pushed in. I released the key, hit Stop and Restore together, and 128 mode worked perfectly. What a scare!

*John Mahowald
Address Unknown*

○△☆ **Reviving Dead Computers**

I killed my 64 by shorting the wrong contacts in an abortive attempt to reset it. Fearing the worst, I was mentally prepared for a \$100 repair bill and a long absence from the joys of computing. But because I'm a decent electronics tinkerer, I opened up my computer to see what a dead 64 looks like. To my great surprise, I found a glass fuse inside, obviously blown. I went to my local electronics store and got a 1.5 amp 250 volt fast-acting fuse, which I substituted for the dead one. Needless to say, the computer came nicely back to life.

*Chris Heller
Starkville, Mississippi*



Power Supply Problems

A frequent cause of computer trouble is some sort of failure inside the external power supply. Some of these supplies are easier to repair than others, but you can always try something if you suspect that yours is defective.

The first thing to do is to isolate the problem to the power supply. The surest way to do this is by borrowing a friend's good power supply and temporarily switching it with yours. If your symptoms continue when the new power supply is connected, the underlying problem is elsewhere in your system. If the symptoms go away, you know that your own power supply is defective. That probably means the fuse is blown.

Because Commodore power supplies come in many styles from several different vendors, there's no one way to tell you how to repair them. But with these general instructions and your own skill and common sense, you should be able to get by.

Before working on your power supply, or on any piece of electrical equipment, you must be sure that it's unplugged. With the blades of the power plug in sight, examine the power supply to see if you can find the fuse, or if the power supply can be disassembled.

In some units, the fuse holder is under a circular black cap that can be removed with a screwdriver. Others have four screws that can be removed to let you open the box. Sometimes these are covered with plastic plugs which look like part of the cover. You can remove them with a knife or other pointed instrument.

If you find the screws, remove them and carefully disassemble the supply. When it's open, check the glass fuse inside. Usually a visual examination can tell if the fuse is blown, but only an ohmmeter test can tell for sure. If the fuse is blown, replace it with one of identical rating. Usually these fuses are of the 3AG type, rated at 1 or 1.6 ampere. Replacements are available at Radio Shack and elsewhere; you might want to try the Slow-Blow type, but you should never use a fuse with a higher ampere rating.

If your power supply is of the so-called potted type, where everything is sealed in black plastic potting compound, you still have a chance of replacing the fuse. Pry off the bottom cover, and put the power supply in front of you on the bench, upside down with the power cord going off to your left. Measure in from the left $1 \frac{5}{8}$ -inches and up from the bottom (closest to you) $1 \frac{1}{4}$ -inches. Chip away the potting compound in this area, and you'll find a fuse buried $\frac{1}{8}$ -inch to $\frac{1}{4}$ -inch deep. Chip it clear, then replace it with a new fuse as described above.

Clean off the ends of the new fuse to the bare brass, then solder it rapidly to prevent oxidation.

If your fuse is good, and if your power supply can be disassembled, there are still some things you can do. Solder joints can go bad, especially around the large capacitor, although they may look fine on the surface. Simply remelt all joints with a soldering iron, and you may be up and running.

Finally, you can replace the voltage regulator with an ECG 1934 or equivalent, available at many electronics stores for about \$8.00. If that fix doesn't work, resign yourself to paying for a new power supply.

*Bruce Jaeger
St. Paul, Minnesota*



Dead 128

If your 128 behaves as though it is dead, you've likely blown one of the two fuses in the power supply. If the computer doesn't do anything when you turn it on, the culprit may be the 4A 125V fuse inside the power supply. Be sure to unplug everything before disassembling the unit, and be careful what you touch—the internal capacitors can hold a pretty shocking charge! If you turn on the computer and the drive and printer initialize but nothing else works, replace the 1.6A 250V fuse which is easily accessible from the bottom of the power supply.

*Al White
Lufkin, Texas*



Schematic Diagrams

The programmers reference guides for the 64 and 128 contain schematic diagrams for their respective computers. You can use this information to narrow your search for trouble.

If you want even more thorough service data, it's available in the Computerfacts series of publications from Howard W. Sams. A Computerfacts folder includes schematics, troubleshooting tips, preliminary service checks, replacement parts cross reference lists, and other similar information.

Computerfacts are available for the 64, 128, Plus/4, VIC20, Amiga 500 and their peripherals. They cost from \$20-\$40, depending on their size and complexity. To order or to find a local dealer, call Sams at (800) 428-7267.

Related Tips in Other Chapters

Here are some other tips about troubleshooting:

Chapter 9	Troubleshooting Joysticks
Chapter 12	Head Stick Fix Disk Head Unsticker Avoiding Loading Slowdown
Chapter 16	Cartridge Tips Cartridge Lifesaver
Chapter 17	SX 64 Keyboard Repairs

A Hardware Assortment

“I submit to the public a small machine of my own invention, by means of which you alone may, without any effort, perform all the operations of arithmetic...”

Blaise Pascal
Hardware Advertisement
1644

This chapter contains hardware-related treasures that don't quite fit in elsewhere. They involve Reset buttons, the serial bus, cartridges, construction projects, and other similar topics.



All About the Reset Button

Here's a summary of the important features of this sometimes mysterious control. The information presented is based on years of personal experimentation with many different computers. Due to periodic ROM changes, your own machine may perform a bit differently.

The 128 comes with a Reset button on the right side of the computer, next to the main power switch; the Plus/4 has a Reset button, too.

The 64 comes without a Reset button, but many 64 owners have installed one, often by plugging it into an external port. There are three common connection points on the 64:

- 1) Between pins 2 and 6 of the serial I/O port. Get a plug to fit the port, and connect a normally open pushbutton between these two pins. Identify the proper pins by looking for the tiny numbers molded into the plastic part of the connector. Be careful if you use published diagrams for the connectors, because they don't always say which end of which sex connector they depict. Diagrams in Commodore manuals usually show the solder terminal end of the male connector.

- 2) Between pins 1 and 3 of the User I/O port. Follow the instructions above. Be careful, because it's easy to insert your connector upside down, putting your switch across the wrong two pins. If you can get a polarizing key for your connector, you can avoid that possibility.

- 3) Between pins A and C of the Expansion Slot. Be extra careful that you use the proper pins, because Commodore diagrams sometimes identify them differently from the industry standard. Many people use a discarded cartridge to house this type of Reset button.

When you press the Reset button, the computer behaves almost as though you turned it off and then back on. We say "almost" because turning the power off erases everything in memory, while using the reset button only resets part of memory. Using the Reset button also saves wear and tear on the power switch and many internal components.

Essentially, pressing the Reset button activates the routines that set the computer's memory to its "power up" condition. Locations below the start of BASIC are initialized, as are the I/O locations in high memory. Zeroes are put into the three lowest positions in BASIC's user area, but other memory locations are left alone.

Here are the most important effects of this process. Although they are specific to the 64 and 128, most of them also apply to the Plus/4:

1) The computer is removed from any lockups, endless loops, or other undesirable states which may exist.

2) All peripherals on the serial bus are reset. This often corrects lockups and other problems having to do with the disk drive, printer and modem.

3) Memory locations below the BASIC program area are restored to their power-up values. This disables any machine language programs that might be in use, even though the programs themselves may still remain in memory. It also clears the cassette buffer, erasing any machine language stored there, and it resets the 128's function keys to their power-up definitions.

4) All memory-mapped I/O locations are returned to their power-up values. This resets screen colors, special graphics modes, the SID (sound) chip, etc.

5) Your BASIC program remains in memory, but because pointers have been reset and the zeroes have been inserted, the computer cannot find it. Later on, we'll show how to resurrect the program on the 128. On the 64, you will have to use an UNNEW program.

6) Machine language in the BASIC program area is untouched, as is any in the 64's protected area from 49152-53247 decimal.

7) The 128's high-resolution graphics screen is preserved, except that the first three bytes are changed to zeroes and the GRAPHIC 0 mode is invoked.

On the 128, several other keys, if depressed when the Reset button is pushed, will modify the computer's response in very important ways:

If the Commodore key is held down when the Reset button is pressed, the computer will go into 64 mode, and any programs in memory will be lost. If you have a 1571 disk drive, it will recognize only the first side of any disk it may contain. You can use both sides in 64 mode by resetting in 128 mode, then entering GO 64 and responding YES to the prompt.

When entering 64 mode by using the Reset button, be sure to keep the Commodore key depressed until you see the 64-style message on the blue screen. If you let it up too soon, you could return to 128 mode.

If the Stop key is held down when you press the Reset button, the computer will be reset to 128 mode, but with the Machine Language Monitor active. BASIC is not initialized, so if there was a 128 BASIC program in memory at the time of the reset, it will still be accessible

at this time. To find it, enter X and press the Return key to exit the Monitor, then do a LIST to see your program. As you can see, using Stop with the Reset button can be very useful for uncrashing BASIC programs in 128 mode.

Even though it doesn't disturb BASIC, this method, like the others, returns the function keys to their power-up values.

Surprisingly, the Stop/Reset combination also calls the Monitor from 64 mode, although it's not 100% useful. You can use the Monitor to examine or change 64 memory locations, and even to save blocks of memory to disk or tape. But you can't use the X command to exit from the Monitor, so you're stuck there until you use the Reset button again.



128 Reset Button Extender

My fingers are large, and I find the tiny Reset button almost impossible to use, even with my pinky. So I took a 3/4-inch piece of Radio Shack shrink tubing, slid it over the button, and used my wife's hair dryer to heat it for about ten seconds. This was long enough to shrink the tubing over the button, and left about 3/8-inch of tubing exposed. Now I can reset the computer with my index finger.

*Walt Grosch, KZ9F
Milwaukee, Wisconsin*



Resetting the 128, Plus/4 and C16

If you have a program crash, the Reset button on these computers comes in very handy. Hold down RUN/STOP while depressing this button, and the computer will go into the Machine Language Monitor without disturbing anything in memory. To get back to BASIC, type an X then press Return. Unless your crash was particularly disastrous, your BASIC program will be intact.

*Norman Bungay
Bay L'Argent, Newfoundland*



Disconnection Discombobulation

Have you encountered those commercial programs that won't run unless everything is disconnected from the serial bus? Every time you want to run one, you have to disconnect your printer, your second disk drive, and anything else you might have connected to the computer.

The only thing worse than disconnecting all that equipment is listening to the people who say they don't have to disconnect it, even with the same program that you have.

When my computer was in the shop last month, I discovered something that might explain the discrepancy. I use a 128 with a 1571 drive, and I routinely have a second disk drive (a 1541) and an MPS801 printer connected to my serial bus. I'm a frequent user of Quantum Link (in 64 mode), and although I had heard of the need to disconnect everything before loading the Q-Link software, I had never had to disconnect anything in my own setup. I attributed all the "disconnection advice" to people who didn't know what they were talking about.

But when my 128 and 1571 went into the shop, I replaced them temporarily with a 64 and a 1541, leaving my printer attached to the serial bus. Every time I attempted to access Q-Link, my system hung up while loading the software. A call to Customer Service yielded the advice to "disconnect the printer." Given my previous experience, I followed this advice with reluctance. Imagine my surprise when it worked! Once I unplugged that printer cable, the software behaved perfectly. Whenever I plugged it back in, the previous problem returned.

When my other equipment returned from the shop, I connected everything up (including a second drive and the printer), and had no further trouble loading the software.

The moral of the story seems to be this: When the software people say to disconnect those cables, they probably know what they're talking about. But depending on your equipment, it might not be required. So try things both ways and see how much disconnecting you really have to do.

*Lil Willard
Marietta, Ohio*



Beating the Serial Bus Blues

My Quantum Link software, like many commercial programs, won't work if a printer or other peripherals are connected to the serial bus. Just turning them off isn't enough—they have to be physically unplugged from the bus.

After months of moving my drive to get to the back panel, and months of wear and tear on my fragile connectors, I decided to do something better. A little experimentation showed that the connection causing the problem was the ATN line at pin 3 of the serial bus plug. This is the pin directly opposite the guide notch.

I made up a short connector cable with a miniature SPST toggle switch in this line, and now I can use my software without unplugging

everything. My local Radio Shack carried all the parts, and I built the cable in a few hours for just a few dollars.

*Bruce H. McMillan
East Brookfield, Massachusetts*



Cartridge Tips

If one of your plug-in cartridges stops working, try it out on another computer. If it works O.K. there, it may mean that your power supply is on its way to failure, and doesn't have enough oomph to drive the computer and the cartridge together. It might also mean that your computer's cartridge port connector is dirty; you can often clean this problem up with a pencil eraser.

If you have a cartridge expansion board, it probably has a switch for disabling any of the cartridges inserted into it. Sometimes you can look into the cartridge ROM by powering up with the cartridge in place, then using the switch to disconnect the cartridge. BASIC's PEEK function can then often be used to examine the cartridge.

If you're capable of working on PC boards, you can install your own cartridge on-off switch by using a double pole switch to break the lines going to pins 8 and 9 of the cartridge port. To disable the cartridge, just turn the switch OFF. To enable it, turn the switch ON and reset the computer.

If your cartridge uses one or more SYS or other commands, it's useful to write them on an adhesive label and fasten it to the top of the cartridge where you can refer to it as needed.

*D. L. Jassby
Princeton, New Jersey*



Use a Cartridge Expander

I frequently use the Simon's Basic cartridge with my 64, but I often program in the original BASIC that comes with the computer. When I became worried about the effects of constant insertion and removal of the cartridge, I bought a cartridge expander with three slots, a selector switch, and a Reset button.

Now all the wear and tear is on the sockets in the cartridge expander.

*Frantz Parisot
Port-au-Prince, Haiti*

○△☆ **Cartridge Lifesaver**

Sometimes after heavy use, a cartridge will fit loosely in its slot, and will tend to disconnect when the keyboard is bumped. If this happens to you, don't throw the cartridge away!

Instead, go to a sewing store and buy two feet of thin elastic. Put your cartridge in place, then wrap the elastic around the computer's body and the cartridge, allowing the elastic to lay between the function keys and the Restore key. Use a safety pin to make the elastic into a snug loop, then enjoy your cartridge for a few more years. When you aren't using a cartridge, you can remove the elastic or leave it in place.

*Walt Latocha
Oak Park, Illinois*

○△☆ **RAM Expander RAMDOS**

If you have a RAM expansion unit for your 128, be certain you have the RAMDOS program, which allows you to use your expander as a very fast disk drive. Many early RAM expander units were shipped without this valuable program, which is now available on Q-Link and in many user group libraries.

*Luke A. Babey
Port Washington, New York*

○△☆ **Become a Hardware Hacker**

In *Commodore Magazine* and elsewhere, you'll sometimes find articles about build-it-yourself accessories for the User port. These little gadgets let your computer control lights and switches, and/or sense information from the outside world.

Don't be afraid to try these amazing projects—completing one is one of computerdom's biggest thrills! I very well remember my PET User port projects of 1980-1981. After building the first ones, I figured out how to design them. Along the way, I mastered some of the most important pieces of microcomputer knowledge, and got my start as a Commodore "expert."

Electronics construction is a fascinating field, and with today's components it's easy to build a project that works. The parts for most projects can be bought at your local Radio Shack, and the skills for assembling them are not very hard to acquire. If you can use a screwdriver, you can learn to use wire strippers and needle-nose pliers. Learning to

solder is a little more difficult, but anyone can become proficient with an hour or so of practice.

Learning to program the User port is like learning to ride a bicycle—a challenge for the learner, but very well worth mastering. To really understand it, you have to get in-depth knowledge of PEEKs, POKEs, binary and hexadecimal. That isn't very much to learn, but for most of us it's knowledge of a new and different kind, so it seems to be very difficult. But once your decimal-trained brain has learned to think in binary, you'll be amazed how simple it is.

And when your first construction project actually works, you'll remember the feeling forever.



Joystick Pieces

Many construction projects call for a nine-pin subminiature connector to fit into the joystick port. A convenient and economical source for such connectors is an old Atari, Commodore or similar type joystick that has been taken out of service. These can often be found at yard sales, flea markets and liquidation sales, and the price is usually right.

Not only are they inexpensive, but they include a molded cable with solderless lugs on the other end. Just take the joystick apart and pull the lugs from the circuit board. If you believe in saving everything, no matter how small, you can cut off the joystick's rubber feet and use them for other projects around the house and computer room.

*Matthew Hately
Bowmanville, Ontario
Canada*



Burning In

Modern electronic equipment is extremely reliable, and for many of us service calls are a thing of the past. In fact, if a newly manufactured piece of equipment is going to fail at all during its life, it will probably fail during its first few hours of operation. Where reliability is important, it's common to "burn in" new equipment by letting it run for several days, with periodic operational checks.

You can burn in the new equipment you buy or build, with several advantages to you. All you have to do is turn the equipment on, and leave it on all day and all night. It's desirable, but not necessary, to have it doing some innocuous task. To be on the safe side, somebody should be there with it if possible, in case something starts to smoke.

From time to time, put the equipment through its paces to see if anything has failed. If it comes with a diagnostic program, run it. If not, do something that exercises the major features. When the burn in period is finished, repeat these tests once again.

The length of the burn in period is up to you, but the longer the burn in, the better. It should be something between a few hours and two or three days or more. Remember, you want to give the equipment every chance to fail during its critical first hours of operation.

Here are the advantages of burning in. First, you put in enough hours during the warranty period to decrease the chance of later out-of-warranty failure. Second, if the product fails so soon you can take it back to the retailer while he still remembers your purchase. And third, you get a head start on learning your equipment and its quirks.

*Cecil C. Serfent
Chesapeake Beach, Virginia*

Related Tips in Other Chapters

These tips also relate to the hardware topics that were covered in this chapter:

Chapter 12	<i>ρ93</i>	Easy Disk Drive Reset
Chapter 20		128 Machine Language Monitor and Resets
Chapter 22		128 CP/M Reboot
Chapter 23		128 Reset Changer
Chapter 33		Commercial Software and the Reset Button

Other Computers

“Half the world knows not how the other half lives.”

George Herbert
Jacula Prudentum
1651

Because they are the world's most numerous computers, Commodore's 64 and 128 are the focus of most of the tips in this book. But Commodore makes other computers, and believe it or not, other companies make computers, too! This chapter focuses on the Amiga, Plus/4, 16 and SX 64, and on the IBM PC/XT/AT and clones. The tips are alphabetical by type of machine.



Generic Tips and Tricks

Many of the tips in this book will work with any Commodore computer. Many will work with any computer at all. If you're in doubt about a given tip, try it and see if it works.

The commercial software tips are a fertile field for this activity, because many commercial programs come in versions for several computers. The chances are surprisingly good that a 64 or 128 tip for one of these programs will work on another machine.

Amiga Border Remover

Sometimes, when I write a program, I want to eliminate the workbench border containing the "gadgets." To do it, I make the window with an empty title bar, then set palette 0, palette 1 and palette 2 equal to each other. This easily gets rid of the border.

*Jesse Peterson
Donahue, Iowa*

Amiga Default Colors

When working in Amiga BASIC, sometimes the system's default colors are changed to an undesirable combination. You can return to the original defaults by stopping the program then doing this: press the right Amiga key and the R key simultaneously, then quickly press the right Amiga key and the period key. If it doesn't work the first time, try it again, this time more quickly.

*Jesse Peterson
Donahue, Iowa*



IBM/Commodore Compatibility

Many Commodore owners have access to some sort of IBM PC or clone, or can see such access in their future. I recently added a clone to my computer room, and have found some products that make it much easier to work with the combined equipment.

The first is Omnitronix's IBM TO CBM Printer Port Adapter. This cable and software convert the PC's parallel printer port to a Commodore-compatible printer port. With it, I can use my Commodore DPS1101 daisy wheel and MPS801 dot matrix printers with the new PC. Think how much money I saved by not buying two new printers!

Both printers work fine with my clone, and there's very rarely a hangup. I can list programs, dump text screens, and do all the other common printer tasks, but I haven't attempted graphics. My word processor drives the DPS1101 without any problems.

The IBM TO CBM is made by Omnitronix, 760 Harrison Street, Seattle, WA 98109. It's available from them or from their dealers.

Another valuable product is Big Blue Reader 128/64, a program that lets the 1571 or 1581 read and create MS-DOS compatible disks. (MS-DOS is the PC's disk handling system). Unfortunately for those who don't own a 1571 or 1581, the Big Blue Reader will not work with the 1541.

Using Big Blue Reader, I'm able to move my Easy Script files and my Microsoft Multiplan spreadsheets over to my clone, where they can be processed by the more powerful software. I'm also able to move files from the clone to the 128.

Big Blue Reader is available from dealers and from SOGWAP Software, 115 Belmont Road, Decatur, Indiana, 46733.

If you plan to get a PC in the future, you might want to start buying double-sided double-density disks right now. They will work with all Commodore drives, are the preferred disk for the 1571 in 128 mode, and can be easily reformatted to work with MS-DOS. They're often only pennies more costly than SSDD's.

Although I haven't tried it myself, it's been reported that the 1902 Monitor is compatible with the PC's Color Graphics Adapter (CGA) mode. However, it's not compatible with monochrome or EGA modes. Commodore's new 2002 may be even more compatible. If you own one of these monitors, look into compatibility before acquiring your PC; you could save hundreds of dollars by avoiding a monitor purchase.



Notes from PC Land

Here's what I've observed about my new PC clone, as compared to my long-familiar PET/VIC/64/Plus4/128 machines:

- The PC is much less fun. Commodore-land has lots more interesting programs and lots more interesting programmers.
- Though my own machine is a \$3000 AT clone, Commodore's Colt and other PC's look very, very nice, and will fill most people's needs with lots of power to spare.
- The hard disk drive makes loading and saving instantaneous. Other than the price, that's the biggest tangible difference in the systems.

- Another big difference is filenames. MS-DOS is limited to eight very specific characters, while we get to work with sixteen of anything we want.
- The difference in the keyboard layout is driving me insane, but I do like having four separate cursor keys laid out in an easy-to-finger “T” pattern.
- The 80-column EGA display is great. The CGA isn’t, but it’s hundreds of dollars cheaper.
- There’s a silly little synthesizer, with one monotonous voice. No harpsichords here, folks!
- Programs that require 256K on the PC are definitely not four times better than those that fit comfortably into 64K on the Commodore machines.
- They certainly are four times more expensive, and usually more than that.
- The BASIC takes getting used to. There are many new statements and commands, and many of the familiar ones do unfamiliar things.
- BASIC graphics are similar to those on the 128, but there aren’t any sprites.
- It’s much harder to work with screen and character colors in the PC’s version of BASIC.
- The technical documentation is overwhelming and opaque. The non-technical is oversimplified and a pain.
- The magazines are full of ads and product reviews, but relatively empty of type-in programs and the sorts of goodies you find in this book and the monthly columns which spawned it.

○ Plus/4 Resources

Some Plus/4 owners think they’ve been deserted and their machine declared an orphan. But thanks to the cohesiveness of computer owners, neither of those worries is justified. The 1000-member Plus/4 Users’ Group continues to be a clearinghouse for Plus/4 information, hardware and software. Membership costs \$20 per year, and buys an eight-issue subscription to PLUG, their professionally written newsletter.

A recent issue of PLUG ran to 12 pages, including a resource directory, software library listings, letters from members, and classified ads.

Members may also use the PLUG disk library for a minimal copying and postage charge. The library has about 25 disks full of various

programs for the Plus/4. One disk I looked at contained a Morse code training program from a member in Germany, a conversion of the 64's animated "Max Headroom graphic," a program that lists and sorts all the amendments to the U.S. Constitution, some information about PEEKs, POKEs and screen codes, and almost two dozen other items.

To join, send \$20 (\$25 outside US/Canada/Mexico) to PLUG, Box 1001, Monterey, CA 93942, U.S.A. Do the overworked staff a favor: don't write for information about joining; just join. The information you get will be worth many times your modest investment.

Calvin Demmon is editor of the newsletter and contact person for the group. You can reach him by mail at the above address, or on CompuServe at 73537,404. If you use the mail and request a response, please enclose a self-addressed stamped envelope.

○ **Plus/4 Documentation**

There's a resurgence in interest in this machine, which was one of Commodore's under-appreciated gems. Its manual is one of the best for any computer, but it has been hard to find additional documentation or other support. Well there's plenty of it out there, if only you know where to look. Here are some of the places.

"The Programmer's Reference Guide for the Commodore Plus/4" by Cyndie Merten and Sarah Meyer. 439 pages, \$21.95, published by Scott Foresman. Invaluable resource of hard-to-find information, including programming tricks and techniques.

"The Commodore Plus/4 Book" by Sarah C. Meyer. 295 pages, \$16.95, published by Howard W. Sams. For beginners only.

"Commodore Plus/4: Computerfacts CC9" also from Howard W. Sams. 21-page packet of schematics and technical data. Definitely not for beginners, but a valuable resource. \$21.95.

"The Complete Commodore Inner Space Anthology" by Karl J. H. Hildon. 122 information-packed pages of data and memory maps on all Commodore computers, including the Plus/4. Published at \$12.95 by *Transactor* magazine.

"Machine Language for the Commodore 64 and Other Commodore Computers" by Jim Butterfield. Contains memory maps and information on the built-in Machine Language Monitor. \$16.95 from Brady Communications Co. 326 pages.

Calvin Demmon
Monterey, California

○ **Plus/4 Items in This Book**

About half of this book's tips and tricks are usable with the Plus/4 computer. Some are usable as is, and some require slight changes. You can identify these Plus/4-compatible items by the circle in their title.

○ **Plus/4 Machine Language Storage**

Memory locations 1630-1771 were designed for speech software. Because the area is almost never used for that purpose, it's a good place to put short machine language programs. It's protected and hidden from BASIC so you don't have to manipulate any zero-page pointers to protect it, and using it doesn't steal any memory from the BASIC program area.

Be careful not to go past location 1771, because the BASIC stack begins there.

*Michael Kaszubski
Baltimore, Maryland*

○ **Plus/4 Graffiti**

“Fool's names and fool's faces are always found in public places,” my mother used to say—trying to discourage us from marking up fences, trees and public walls. She never said anything about names in private places.

Type SYS 52651 and press Return to see the names of some Plus/4 designers. See other names by entering MONITOR, then M BC20 to get a dump of a small area of memory.

*Alfred E. Neuman
Potrzebie, New York*

○ **Plus/4 and C16 Line Width**

These computers lack the 128's WIDTH statement to get normal or double width lines when drawing lines, circles and boxes. But you can double the width of the lines by using POKE 743,1. To return to normal, use POKE 743,0.

*Cyrus Nasiraey
Meshed, Iran*

△ **SX 64 Keyboard Repairs**

If some of your SX 64 keys stop working, gently pry the top off the keyboard. Undo the screws to expose the keyboard connector, and look

for one or two wires that seem to have slipped out of place. When you return them to their normal positions, you may have corrected the problem.

*David Bradley
Address Unknown*



SX 64 and Printer Interfaces

The popular SX 64 computer has no Cassette port, so there's no obvious way to pick up power for printer interfaces and other units that come with a Cassette port connector. But never fear—there are many other places to pick up your + 5 volts.

The first is the User port. Pin 2 of this port is a source of + 5 volts, suitable for your interface. If you have an edge connector to fit the User port, just wire it to your interface. If you can be satisfied with a less elegant solution, take a hacksaw to the Cassette port connector on the interface, and make it fit the end of the User port. Never plug it in backwards, and you should be able to use it without trouble.

Pin 7 of the joystick port is another convenient source of + 5 volts, and suitable connectors are available at Radio Shack.

Finally, you can use pins 2 and 3 of the Cartridge Port. Dismantle an old cartridge, connect your interface to pins 2 and/or 3, and you're in business once again.

Although these three places are convenient sources of + 5 volts on the SX 64, you should be careful how much current you attempt to draw from them. The built-in power supply is not very hefty, and if you tax it very heavily, you run the risk of blowing it. If your printer has its + 5 volt power supply connected to pin 18, as many of them do, you should consider drawing your power from that source instead of the SX 64. Instructions for doing that can be found in Chapter Seven's "Power for Your Interface" tip.

*Itty Greer
Glenshaw, Pennsylvania*

Related Tips in Other Chapters

This tip is of interest to owners of the Commodore PET computer:

Chapter 28 64 to PET Translation

Learning to Program

“A man may write at any time, if he will set himself doggedly to it.”

Samuel Johnson
March, 1750

Everyone knows that computing’s ultimate pleasure comes from writing a program from scratch. So if you aren’t yet a programmer, becoming one is never too far from your mind. Because you have the computer, and because you have the desire, why not learn how to program? This chapter can help you get started.



The Way We Were

In the early days of personal computing, every computer owner needed to program. Because commercial software was nonexistent, if you wanted your system to do something, you had to program it yourself.

There were almost no programming courses, and very few magazines or user-friendly books, so most of us became self-made men, at least where programming was concerned.

Everyone learned BASIC right away, usually from a book featuring timesharing versions of the language, with nothing like the flexibility of the Commodore dialect. Each month the newsletters revealed new wonders of Commodore BASIC, and we waited for every issue so we could expand our computer horizons.

In those days a newly-discovered PEEK or POKE was cause for international excitement, as programmers around the world incorporated it into their creations.

As our collective knowledge advanced, BASIC no longer satisfied our need for increasingly better programs. Many of us got a monitor program and began wrestling with machine language. ML was difficult, strange and unusual to us self-taught BASIC experts, but with practice and perseverance we learned to make it work.

One of the most popular techniques, still used widely today, was to combine BASIC programs and short machine language routines. This kept the simplicity of self-documenting easily modifiable BASIC, while adding the speed and transparency of machine language.

Pretty soon assembler programs became available, making machine language much easier to create, modify and document for the enlightenment of others. Those with the patience for such things wrote wonderfully powerful programs, and those without it used them. The programs got better and better, increasingly being produced by full-time machine language specialists.

And slowly but surely, the need to program diminished. One day, no one knows exactly when, the commercial programs became so good and so widely available that people could use them exclusively, never having to learn to program.

That's the situation today, and I don't know if it's better or worse. I do know that computerists who can't program are missing one of the most rewarding activities in computing. But I also know that the commercially-available programs—the games, the word processors and all the other miracles—are much better than anything one person is likely to write himself.



Why You Should Learn BASIC Programming

As you know from your own experience, you can get plenty of use out of your computer if you don't know a PRINT statement from a FOR...NEXT loop. But to live computer life to the fullest, you have to learn to program.

For most of us, that means learning to program in BASIC. After all, it comes free with your machine, and it's the most widely-spoken programming language in the world.

If you know how to program in BASIC, you're among the computer elite. Of course there are more powerful languages, and in the circles that use them BASIC is often looked down on, but think a little further. Of all the people using computers, how many of them know how to program? Not very many at all. So the lowliest BASIC programmer stands head and shoulders above the crowd of common computerists. Interesting position, isn't it?

Not only does learning BASIC make you a programmer, but it pays other dividends as well.

Once you know how to program, you can make your computer do things that your commercial programs cannot. It's like being a do-it-yourself carpenter — if you need a shelf or a storage shed, you can make it with your own two hands. And if you need one that's not available in stores, you can fabricate it to your unique requirements. So when your commercial software can't quite do what you want, there's a good chance you can write something that can.

Even if you never dabble in do-it-yourself programming, your knowledge of BASIC empowers you in working with commercial programs. In my own situation, for example, knowing BASIC helps me get the most out of at least seven commercial programs: the Microsoft Word word processor, the Superbase and Paradox database programs, the Multiplan and Lotus spreadsheets, the Bitcom communications package, and the MS-DOS operating system.

Each of those programs includes its own small programming language, put there so users can customize them to their own needs. Each of those languages has BASIC-like features that would be very difficult for anyone who didn't know programming in BASIC.

Because I know Commodore BASIC, I'm better with Lotus 1-2-3. Wow.



How to Learn BASIC

Here are several time-proven tips on learning to program in BASIC. If you follow them, your journey to mastery will be shorter and more pleasant:

1) Don't think BASIC is complicated. The BASIC on the 64 has under 80 keywords, and half of them are very seldom used. The 128's BASIC 7.0 has about twice as many. Compared to the vocabulary of any human language, either number is ridiculously small.

2) Get a reference book. The Programmer's Reference Guide for the 64 and the System Guide for the 128 have comprehensive descriptions of the syntax and action of every BASIC keyword. Because programming success depends on taking advantage of these keywords, you should have the complete definitions on hand.

3) Get a textbook. By this I mean a book whose purpose is to teach you programming in BASIC. While a reference book describes the keywords in alphabetical order, a textbook explains them one by one, usually proceeding from the simplest to the most complex. It includes sample programs, and usually has quiz questions and work assignments as well. You'll find books like this in any well-stocked bookstore, especially those at colleges. It's nice to get one oriented toward Commodore computers, but BASIC has enough similarity from machine to machine that that isn't absolutely necessary.

4) Think about taking a course. There are thousands of BASIC programming courses available, for credit and for fun. Taking one gives you the discipline to move through the learning process, and it puts you in touch with a teacher and other students with interests similar to yours. If you can't find a course taught on Commodore machines, find one on any other micro, or even on a time-shared mini or mainframe. Don't be concerned about the length of the course, even if it's a full semester long — it will be over before you know it.

If you're on Q-Link, go to the Commodore Software Showcase and download "basic basic.sda", uploaded by LouSander1. It's a self-teaching course in elementary BASIC, and it's been helping Commodore people for years. (Note that on Q-Link, as everywhere, program names are typed in unshifted characters. Also note that on Q-Link, unshifted characters appear as lowercase).

The BASIC BASIC program is also on the companion disk to this book. By itself, it's worth the price of a disk. Page 391 has an order blank.

5) Start modifying simple programs. Learning by doing works wonderfully with BASIC. Find some simple programs and change them to your liking. Add on-screen instructions. Make them user friendly. Embellish the features they already have. You'll not only learn the original programmer's intent, but you'll exercise your own programming skills. Figure 18-1 shows a program for adding a series of numbers; improve it as much as you can.

```
10 REM HOW TO LEARN BASIC - SANDER
20 REM ADDS NUMBERS
30 INPUT A
40 INPUT B
50 INPUT C
60 D=A+B+C
70 PRINT D
```

Fig. 18-1.

6) Remember a very old joke: A man is walking the unfamiliar streets of New York, trying to find the auditorium where his nephew's music recital is about to begin. Worried about being late, he approaches a native New Yorker. "Excuse me sir," he asks, "how do you get to Carnegie Hall?." The New Yorker looks down his nose at him and says haughtily, "Practice, practice, practice!"

The same is true of programming — it takes hours and hours of practice, trying out new things and modifying things that you've tried. The practice is fun, the result is rewarding, and the hours of effort are worth it.



Learning BASIC

Remember the suggestion about downloading the Q-Link program "basic basic.sda", uploaded by LouSander1? Here's an E-Mail message I got from a Q-Linker named ENT.

"Thank you for uploading BASIC BASIC. I have had my computer for over a year, and after doing the first section of BASIC BASIC I wrote my own first program! Thank you again."

Q-Linker OMD1 says "BASIC BASIC fills half a disk side, and it's explaining BASIC much more intelligibly than the 64C System Guide or the Programmer's Reference Guide."

○△☆ **Comparative Anatomy**

Chapter 31's "Directory Viewer" and "Blocks Free Reader" programs have much in common, and you can learn quite a bit by comparing them. They were written independently by different people, but they both read information from the disk directory and make that information available to a BASIC program.

Notice, for example, that only one of the programmers implemented a User Defined Function.

Try to see what is being read from the directory, and try to relate it to the purpose of the program.

By analyzing programs like this, you can increase your skills significantly.

○△☆ **Learning Machine Language**

For most people, machine language is harder to learn than BASIC, at least in the early stages. The problem isn't that machine language is so difficult, but that BASIC is so simple. After all, BASIC was created to make it easy to learn how to program.

The first essential step in learning machine language is to master the binary and hexadecimal numbering systems. That's not as hard as you think, and Chapter 26 can help you.

The next step is to learn the registers, addressing modes and instruction sets of the 6502 series of microprocessors. You can learn this from a book, but for most people it's easier to learn from a teacher. The electronics departments of Community Colleges and technical schools often run elementary courses in machine language programming; computer science departments almost never do.

If you can't find a 6502-oriented course, you might want to try one for another microprocessor. Because there's some similarity among all microprocessor families, and because the hardest part of machine language programming is to learn to think like a microchip, much of your learning will be transferable to the 65xx world.

Once you have that elementary knowledge of machine language, the next step is to work with some actual programs. You'll find dozens of them in Chapter 20's "Digging into Machine Language;" work on them for a while, and soon you'll be writing your own.

Related Tips in Other Chapters

Several other chapters are of particular interest to those who want to learn how to program:

Chapter 1	Learning
Chapter 2	Programs from Books and Magazines
Chapter 20	Machine Language
Chapter 30	Useful Application Programs

These specific tips may also help you:

Chapter 1	Typing in Programs
Chapter 20	Digging into Machine Language

The BASIC Programming Language

“High thoughts must have high language.”

Aristophanes
Frogs
405 B.C.

BASIC is a high-level computer programming language. “BASIC” is an acronym, standing for Beginner’s All-purpose Symbolic Instruction Code. Both were invented by professors at Dartmouth College, who wanted to make programming accessible to intelligent non-specialists. I think they succeeded, don’t you?

By the way, a high-level programming language is one whose words are those of normal English. Low-level languages use numbers and abbreviations approximating the natural language of the machine.

The tips in this chapter refer to specific keywords found in Commodore’s implementation of the BASIC programming language. Basically, they deal with BASIC at its most basic level.

Because most keywords can be used in several versions of BASIC, most of these tip names do not include the name of the computer to which they primarily apply. If you own the computer, you know which keywords are yours. The dingbats will help you, too.

○☆ AUTO Fix

The 128's AUTO command jumps the cursor to the right one position after printing the line number. That's fine for empty screen lines, but not for those containing an old READY prompt. When such a prompt is at the start of a line, AUTO gives something like 80ADY with the cursor flashing on the "D". This requires the user to delete back to the line number before entering his line.

If you reprogram a function key to print DATA, PLAY, or another keyword that you often use at the start of a line, you can avoid the problem by putting a delete character before your statement. A typical example would be:

```
KEY 1, CHR$(20) + "DATA"
```

*J. C. Vollmer
Prior Lake, Minnesota*

☆ Which BANK?

The 128's BANK command is useful for switching memory configurations. Unfortunately, there's no related function to determine the bank with which BASIC is working. You can find the current bank, though, by PEEKing location 981. The number returned is the number that was given in the most recent BANK command.

*Charles Lavin
Coral Gables, Florida*

△☆ Using BEGIN and ELSE

When using IF...THEN...ELSE and BEGIN...BEND in BASIC 7.0, the ELSE BEGIN clause must be in the same program line as the BEND, like this:

```
260 BEND : ELSE BEGIN
```

If the ELSE is on the next line instead, the statement(s) following it will always be executed.

I pass this on because I learned it the hard way, and it was a difficult problem to solve.

*Steven Gollery
Hemet, California*

 ☆ **CHAR Trickery**

To gain access to upper and lowercase characters when using the 128's CHAR statement on a bit-mapped screen, just POKE 4588,216. The normal content of this location is 208, which gives you access to uppercase characters and the keyboard graphics set.

By the way, the CHAR statement works on normal text screens as well as on graphics screens, and can be used to place text wherever you'd like. It's often easier to use CHAR than to use the usual combination of PRINTs, TABs and cursor control characters.

*Dick Johnson
Norco, California*

 △ **CONT for Endless Loops**

The CONT statement is usually used in direct mode to resume program execution after a STOP or END statement. But it can also be used in a program in 64 mode to create an endless loop. Whenever your program executes a CONT, it will hang up until the STOP key is pressed. This trick does not work in 128 mode.

*William Greer
Glenshaw, Pennsylvania*

 ○△☆ **Shortening DATA Statements**

Instead of entering a zero in a numeric DATA statement, it's perfectly acceptable to leave that data item blank! For example, these two lines are exactly equivalent:

```
100 DATA 7,15,39,0,47
110 DATA 7,15,39,,47
```

Leaving out the zero is also said to speed up the READING of the DATA statement.

In string DATA statements, it's usually acceptable to leave out the quotation marks. If you leave out everything (as in the numeric example above), the data item will be read as a null string. For practical purposes, these two lines are identical:

```
120 DATA "LOUIS","BILL","", "TRACEY"
130 DATA LOUIS,BILL,,TRACEY
```

You must use quotation marks if your data items contain shifted characters, commas or colons. Unless they are enclosed in quotes, shifted characters will list improperly and give erratic results when read. Commas and colons will be interpreted as separators rather than as actual data.

*Deglet Noor
Indio, California*



DLOAD and Other Disk Commands

Many of the BASIC 7.0 disk commands have parameters for Device Number and Unit Number. The Device Number is the same as the secondary address of the device in question, and is set by a switch or jumper on the device itself. It is normally set to 8 for the first 1541 or 1571, 9 for the second, and so on. The Unit Number is for those seldom-used devices such as the 4040, which have two disk drives built into one device. For those devices, Unit 0 refers to the first drive, while Unit 1 refers to the second. For devices having only one built-in drive, such as the 1541 and 1571, the Unit Number is always 0.

All the BASIC 7.0 commands use Device 8, Unit 0 as defaults. That is, unless you specify otherwise, the command applies to the first drive on device number 8.

But you can easily specify another drive or unit for the command. These two parameters are usually given after any other required parameters, although they can be given in Unit/Drive or Drive/Unit order. The Drive number is always preceded by a D, and the Unit number is always preceded by a U.

For example, the command DIRECTORY will call up a full directory of drive 0, unit 8, while DIRECTORY,D1,U9 will get a directory for drive 1, unit 9. DIRECTORY U9 will call up a directory for drive 0 on device 9. The D0 can be omitted because drive 0 is the default drive.

The ON clause can be used to skip any parameters preceding the drive or unit you specify. For example, DIRECTORY ON U9 will give the directory for drive 0 on device 9.

*Charles Lavin
Coral Gables, Florida*



DRAW Shortcut

This statement is more versatile than you might think. The single statement:

DRAW 1,160,100,50,50,35,35,190,190

will draw four different points on the screen. Also:

```
DRAW 1,160,100 TO 300,100,50,50 TO 50,100
```

will draw two separate lines, without having to execute a separate DRAW statement for each one.

*IMD Walrus
Quantum Link*



DSAVE with Replace

This 128 and Plus/4 command does have a save with replace option, but it's a little hard to find. The secret is to precede the filename with an @ sign, but without using a colon. `SAVE"@programe"` will save your new version of *programe* and eliminate the old one. Note that the drive number (0) is not needed in the DSAVE command. If you have a dual drive and want to use drive 1, you must use `DSAVE"@programe",D1` or `DSAVE"@programe" ON 1`. On the 64 and other machines, the syntax was `SAVE"@0:programe",8`.

*Charles Lavin
Coral Gables, Florida*



Happy ENDings

Most of the programs I write have a menu, one selection of which sends the program to an END statement. This is fine when the user really wants to quit, but sometimes he has second thoughts, such as when he thinks of something he wishes he had done while the program was running. Doing a RUN at this point is fine, except that it resets all variables to zero and sometimes calls a long initialization routine.

To avoid this problem, I've changed the way I handle END statements. To illustrate, let's assume that the main menu starts in line 500, and that my END statement is in line 1000. My technique is to:

```
1000 END : GOTO 500
```

The program still terminates when line 1000 is executed, but now I have a chance to change my mind. All I have to do is enter CONT, and the GOTO 500 is executed, sending me back to the main menu! It's one of the simplest and most useful techniques I have in my bag of tricks.

*Bob Howe
Tulsa, Oklahoma*

○△☆ FOR...NEXT Hints

The FOR...NEXT loop is one of BASIC's commonest and most useful features. Here are some hints for using it more effectively on Commodore computers.

- Remember that the loop variable is initialized before the loop's end is calculated. You can use this to your benefit, especially when the end value isn't obvious. For example, these two lines are handled identically:

```
100 FOR J=49152 TO 49173 : READ K : POKE J,K : NEXT
110 FOR J=49152 TO J+11 : READ K : POKE J,K : NEXT
```

Line 110 is easier to type, and if you wanted to change the starting point of your POKES, line 100 would require you to recalculate the loop's start and finish, while with line 110 you'd only recalculate its start.

- Your program should never exit a FOR . . . NEXT loop without first completing the loop. If you don't observe this precaution, you'll be plagued with mysterious Out Of Memory Errors.

Figure 19-1 shows some examples of the wrong and right ways to terminate a loop. Notice that when line 140 is executed, J = 5 and the loop is still open. Note how line 210 closes the loop by setting the variable equal to the maximum loop value then executing a NEXT.

```
100 REM WRONG WAY TO QUIT
110 FOR J=1 TO 10 : IF J=5 THEN GOTO 140
120 PRINT J
130 NEXT J
140 PRINT "WRONG WAY TO QUIT!"
150 :
200 REM RIGHT WAY TO QUIT
210 FOR J=1 TO 10 : IF J=5 THEN J=10 : GOTO 230
220 PRINT J
230 NEXT J
240 PRINT "RIGHT WAY TO QUIT!"
```

Fig. 19-1.

- After the loop has finished, the value of the variable is equal to the first value after the loop's end. If the following is executed,

```
100 FOR J=1 TO 5 : NEXT
110 FOR K=6 TO 3 STEP -1 : NEXT
120 PRINT J : PRINT K
```

the printed values for J and K will be 6 and 2, respectively.

Roger Wilcox
Durham, North Carolina



Flashing Cursor for GETKEY

If you are in 80-column mode on your 128, you can get a flashing cursor after your GETKEY prompt by doing this:

```
1Ø PRINT "PRESS A KEY ";CHR$(27);"S";  
2Ø GETKEY A$  
3Ø PRINT A$  
4Ø GOTO 1Ø
```

Don't forget the semicolon at the end of the first PRINT statement.

Andre Leduc
Hawkesbury, Ontario
Canada



Shorter IF Statements

Shorter statements are easier to type and quicker for the computer to process. There are two important ways to shorten most IF statements. Say you have a statement like this:

```
100 IF J<>0 THEN GOTO 150
```

The GOTO is unnecessary. You can always eliminate it after a THEN statement. So it's better to type the above line as:

```
100 IF J<>0 THEN 150
```

The expression $J<>0$ is called a *relational expression*. A relational expression is one using the equals, greater than or less than symbols, or any combination thereof.

When an IF statement is followed by a variable instead of a relational expression, the IF statement will be false only when the variable has a value of zero. It will be true for all other values of the variable.

In our example expression, we only go to line 150 if J is nonzero. Therefore we can shorten it even further to:

```
100 IF J THEN 150
```

James M. Whiteside
Warren, Ohio



INPUT without a Question Mark

When using the INPUT statement for something other than a question, it's desirable to eliminate the "?" that appears after the prompt. There are several ways to do this, but the simplest and most machine-independent is this:

```
100 OPEN 1,0
110 PRINT "ENTER SOME DATA";
120 INPUT#1,A$
130 PRINT
140 CLOSE1
```

Line 100 opens the keyboard as an input device, and line 110 (optional) prints your prompt; don't forget the semicolon at its end. Line 120 puts your keyboard entry into variable A\$. Line 130 moves the cursor to the next line, while line 140 closes the input channel. You can combine the steps into fewer lines if you'd like—we've separated them here to make them easier to understand.

Kaylee Fortney
McCandless, Pennsylvania



INPUTting "Illegal" Characters

By typing quotation marks before and after your response to an INPUT statement, you can include many otherwise unacceptable characters in that response. Commas, colons, leading or trailing spaces, and cursor control characters, all normally illegal or ignored, become part of your input when they are enclosed in quotes. The opening and closing quotation marks will not be included in the string variable you input. In fact, the only character you can't enter this way is the quotation mark itself, CHR\$(34). To input a delete character inside your quotes, press the Insert then Delete keys.

The closing quotation mark, as usual, is optional, unless you need to preserve trailing spaces in your entry.

It's easy to relieve the user of the responsibility of typing the initial quote. Just use the table below to find values for first and second, then plug them into this routine:

```
100 PRINT "YOUR ENTRY";
110 POKE first,34 : POKE second,1
120 INPUT A$
```

Don't forget the semicolon at the end of line 100.

<i>Machine</i>	<i>first</i>	<i>second</i>
C16	1319	239
64	631	198
128	842	208
Plus/4	1319	239

What you're doing here is poking a quotation mark into the keyboard buffer, where it appears as the first character in the user's response.

If you're familiar with such poking, you can make the quotation mark invisible by increasing the number of characters you put into the buffer. Using the three locations starting with *first*, poke the CHR\$(34) equivalent of the screen background color, then poke the CHR\$(34), then the CHR\$(34) equivalent of the character color. Then poke a 3 into *second*, to allow for the extra colors. This version will work on the default colors of a 64:

```
100 PRINT "INVISIBLE QUOTE";
110 POKE 632,31 : POKE 632,34 : POKE 633,154 : POKE 198,3
120 INPUT A$
```

○△☆ INPUT with Default

It's often desirable to have a default value for the variable in an INPUT statement. Here are several ways to provide for that default value. In every case, the user can override the default just by entering a different value.

The simplest default doesn't print on the screen, but is entered as the variable value when the user presses Return. It's accomplished like this:

```
110 X = 3 : INPUT "WHAT VALUE";X
```

When the user presses Return without entering anything else, X will have a value of 3. It works because when Return is pressed by itself, the variable retains the value it had before Return was pressed. If the above statement lacked the X = 3 and the user pressed Return without making an entry, X would remain at zero or whatever other value it had previously been assigned in the program.

A better technique prints the default value to the screen, so the user can see it before he presses Return. Like the first technique, it can only be used when the default is known at the time the program is written. To use it, do something like this:

```
120 INPUT "WHAT NUMBER [SPACE2] 8 [LEFT3]";Y
```


When the cursor appears, it will be flashing over the number 8. If the user presses Return without entering a different number, the computer will accept the 8 as his input.

The above technique can easily accommodate defaults of more than one character, simply by including an additional cursor left for each additional character in the default.

The most sophisticated technique can be used when the default value is not known at the time the program is written, but is calculated somewhere by the program itself.

In the following example, assume that D\$ is the desired default, and that its value has been calculated before line 130 is executed:

```
130 PRINT "WHAT NAME[SPACE2] "; D$
140 PRINT "[UP] "; TAB(9);
150 INPUT N$
```

Line 130 prints the prompt and the calculated default value. Line 140 positions the cursor properly in preparation for the INPUT statement; note that it ends with a semicolon. The number in the TAB statement should be the same as the number of characters in your prompt, not counting the two spaces at the end. Line 150 then accepts the default input, or whatever else the user has chosen to type.

With these three techniques in your bag, you can handle any INPUT default you'll ever need. They all work with either string or numeric variables.



Unwanted Characters in INPUT

If you try to use the INPUT statement with a prompt that is more than one screen line long, the resulting variable will include both your keyboard entry and the prompt!

The way around the problem is to use a PRINT statement to print the first part of your prompt (the part that fills the first screen line), limiting your INPUT statement to the part that overflows to the next line. Because the INPUT statement is now dealing with less than a full screen line, it won't put the prompt into the value of the variable.

On a related subject, don't forget that the INPUT statement accepts everything to the right of the prompt. If you have a border on your screen, INPUT will consider it as part of your response.

Steve Dodd
Edmonton, Alberta
Canada

○△☆ INT Hints

Many BASIC statements and functions automatically perform an INT as an early step in their processing, and in those cases an INT in your program may be unnecessary. Here are some that I've run across in my experiments:

Array Subscripts	MID\$
CHR\$	PEEK
DIM	POKE
GOSUB	RIGHT\$
GOTO	SPC
LEFT\$	TAB

Also, you can often eliminate the need for an INT by using an integer variable, as is shown by the following, in which A% is shown to have a value of 3:

```
100 A%=10/3
110 PRINT A%
```

*A. W. Grym
Oxford, England*

○△☆ INTeresting Bug

If you type `PRINT 8.4*100/2` then press Return, the result will appear as 420. But if you try `PRINT INT(8.4*100/2)` the result will be 419!

Because the computer does all its math in base two, decimal input and output involves several conversions between bases. Because many such conversions involve rounding up or down, the final answer is sometimes not what you thought it would be.

In the case illustrated here, the result inside the parentheses may have come out to 419.99999999+, due to rounding during base conversions. Without the INT, the computer further rounded it off to 420.

With the INT, of course, everything after the decimal point is dropped, leaving you with 419.

*Joe Bian
Brookline, Massachusetts*

○△☆ Double-spaced LIST

When listing programs on your printer, it's often desirable to print them double-spaced. The extra white space makes it easier to follow the program, and it provides a place for notes and possible changes.

You can easily get these double-spaced listings by giving your printer file a number higher than 127. For example:

```
OPEN 128,4 : CMD128 : LIST
```

will produce a double-spaced listing on the printer. When the printing has stopped, reset the printer by executing:

```
PRINT#128 : CLOSE128
```

*Anna Mae Hertzler
Boynton Beach, Florida*



Indented LIST

Some versions of BASIC allow you to put spaces between the line number and the rest of the line, to indent the line when it's listed. This ability can make your listings easier to understand, such as when you indent all the lines contained in a FOR...NEXT loop.

If you put such spaces into a Commodore BASIC line, the computer will automatically remove them, unless you know this trick.

The trick is to type the line number, then any shifted letter, then the number of spaces you want to skip, then the statements you want to have in your program line. When you LIST the line, the shifted letter will have disappeared, but the subsequent spaces will remain. Execution of the program will not be affected in any way.

You can take this one step further to produce a line that is blank except for its number. Type the line number, a shifted letter, a single space, then another shifted letter.

*Stewart Terrier
Alton, Illinois*



LIST Disablers

You can disable the LIST command on the 64 by executing these statements:

```
POKE 774,183 : POKE 775,163
```

To undo the change, execute these:

```
POKE 774,26 : POKE 775,167
```

On the 128, disabling LIST is even easier. Just execute:

```
POKE 774,139
```

To enable the 128 LIST command again, execute:

POKE 774,81

On the Plus/4, these operations can be accomplished by **POKE 775,172** and **POKE 775,139**.

After LIST has been disabled, the program in memory can be run but cannot be listed.

*Jerry Jordan
Middlefield, Ohio*

☆ **128 LIST Bell**

This little trick will make the 128 produce a BELL tone whenever the line is listed. Used cleverly, it can make a program play a one-note tune when it is listed.

- 1) Type the line number and REM
- 2) Type two quotation marks, then CTRL 9.
- 3) Use the cursor keys to backspace to the second quotation mark. Your cursor should be flashing on it at this point.
- 4) Use the INST key to insert two spaces.
- 5) Press SHIFT M, then CTRL G.
- 6) Press RETURN.

*Brian Jacques
Lyn, Ontario
Canada*

☆ **PLAY Statement**

You can use string variables as arguments for the PLAY statement, as illustrated on page 156 of the 128 System Guide. If you want to put more than one variable in a single statement, you can do so by using the plus sign to concatenate them. For example:

10 PLAY A\$: PLAY B\$: PLAY C\$

can be more easily accomplished by:

10 PLAY A\$ + B\$ + C\$

*Jacob M. Smith
Silver Creek, New York*



What are Those POKEs, Anyway?

Many newcomers are mystified by the use of POKEs to disable or change various features of their computers. The POKE magic seems powerful, but arbitrary and unknowable.

Most of the POKEs merely change some memory location in the operating system. If you know about machine language programming, you've probably dug deeply into that subject, so the POKEs are not too mysterious. But if, like most computerists, you're still struggling with BASIC, you're totally in the dark. Here's a brief explanation.

Your computer's operations are very intimately controlled by some built-in software called the *operating system*. This machine language program is permanently stored in ROM (Read Only Memory), where it's activated whenever the computer is powered up. The designers put this critical software in ROM, of course, so computer users and software writers couldn't unwittingly foul it up.

But the designers also knew that software writers would want to change certain parts of the operating system to make the computer more useful with their software. So they set aside certain areas of RAM, (memory that is changeable by the user), to hold these special parts of the operating system. When the computer is powered up, the ROM program writes these parts to RAM, then uses them whenever they're needed.

When you make those mysterious POKEs, such as the ones to disable the LIST command, you're really making minor changes to the RAM part of the operating system. In many cases, you're actually redirecting execution to a new location; it's very similar to changing the line number in a BASIC GOTO statement.



PRINT and the Semicolon

Don't forget that a PRINT statement causes the cursor to skip to the next line, unless it ends with a semicolon. (A comma makes the cursor move to the next print zone, but we're not interested in that here.)

For example, if you clear the screen by PRINTing a CHR\$(147) or SHIFT CLR without a semicolon, the cursor will end up on the second screen line. This can be annoying if you want to print something at the very top of the screen.

To avoid the problem, just end the PRINT statement with a semicolon, like this:

```
PRINT CHR$(147);
```

You can use the same technique when you change the character color. The semicolon will keep your cursor on the same line it started on.

Becky Sue Parton
North Richland Hills, Texas



Structured REMs

This system for using REMs makes it easy to understand the organization of a program, even when the listing is flashing by on the screen. I use it in every long program, making life easier for those who try to decipher my software, and also for myself.

Each program starts with six standard REMs. The first is the name of the program, set off by asterisks before and after it. The second is the date of this version of the program. Then there are four more with my name, address and telephone number. These six lines each occupy 40 columns or less, so no line wraps around on the screen. To make them look even nicer, I insert spaces to center their contents on the monitor.

To end the initial REM section, I put a line consisting of a line number and a single colon. (Believe it or not, this is legal in Commodore BASIC).

I divide the program itself into easily-understood sections. That's usually not very hard, because most programs have several well-delineated functions that lend themselves to such organization.

The first line in each section consists of a REM followed by a space, two or three asterisks and a brief description of the section. I fit this line into 40 columns or less, and I don't use leading spaces to center it.

Like the title section, each of these sections ends with a line made up of a single colon.

Now when the program is listed, its title, date and author are the first things you see on the screen. Its major sections are separated by a colon line followed by a three-asterisk REM line. When you look at one of these listings, it's uncanny how easy they are to follow, either on the screen or on the printer.

To see this REM system in action, download a few of my programs in the Special Edition section of Quantum Link. You'll find it in the Commodore Software Showcase.



RENUMBER Trick

If you have a 128 or Plus/4, you can renumber your programs at the flick of the keyboard. If you have a 64 and one of the many good renumbering utilities, you can do the same and sometimes even more.

But as soon as you remember a program, you lose track of the line numbers of important subroutines or other sections of your program.

If you make up some dummy lines like these, it will be easy to find those routines again:

```
100 GOTO 140
110 GOSUB 700 : REM MAIN MENU
120 GOSUB 800 : REM SORT ROUTINE
130 GOSUB 5000 : REM SOUND BEEPER
140 Rest of program goes here.
```

Because of the GOTO in line 100, lines 110-130 will never be executed. But when the program is renumbered, the line numbers in their GOSUB statements will be renumbered along with the lines themselves. When the renumbering is finished, you can look at these lines to find the new numbers of your valuable routines.

These dummy lines can be anywhere in your program, but at the very beginning is best. If they're there, you can find them in a jiffy. Putting them at the very end of the program keeps them out of the way, but they're a little slower to find.

A. E. Neas
Troy, New York



Disabling SAVE

On the 64, you can disable the SAVE command by executing POKE 818,32. POKE 818,237 enables SAVE again.

On the 128, SAVE is disabled by POKE 818,1 and enabled by POKE 818,78.

On the Plus/4, the commands are POKE 816,136 and POKE 816,164.

Various Contributors
Many Cities



Disabling 64 SAVE

If you want to prevent a 64 program from being saved, all you have to do is adjust the vector in memory locations 818 and 819. To cause a system reset when a save is attempted, do this:

```
50 POKE 818,226 : POKE 819,252
```

You can undo the change by

```
60 POKE 818,237 : POKE 819,245
```

Remember that SAVE will work in the normal way until the first of the above lines is executed.

If you'd like something other than a reset to occur when a save is attempted, that's also easy to do. Just use this generalized routine:

```
50 B = 64738 : HI = INT(B/256) : LO = B - 256*HI :  
POKE 818,LO : POKE 819,HI
```

In the routine, you set B to the address to which you want a SAVE to be directed. Everything else will be automatic. Here, too, you undo your work by executing the POKES in line 60 above.

*Nolan Whitaker
Jeffersonville, Kentucky*



Setting SPRDEF Background Colors

If you've created sprites on the 128, you know how the SPRDEF command works, and how to alter the sprite color by using Control or Commodore and the number keys.

If you want to design your sprite against its planned background color, use the COLOR command before entering SPRDEF. Change the screen color by using COLOR in direct mode, without a line number. For example, COLOR,0,1 will give you a black background. Now enter SPRDEF and enjoy designing your sprite.

*Stephen Knapick
New Rochelle, New York*

Related Tips in Other Chapters

These tips also refer to specific BASIC keywords:

Chapter 5	BASIC Abbreviations
Chapter 23	LIST Stopper
Chapter 27	Debugging Using STOP and CONT Debugging DATA Statements TAB Errors
Chapter 32	64 Remvert

For tips on using BASIC's keywords in programs, see Chapters 23-28. For self-contained BASIC programs, see Chapters 29-32.

Machine Language

*“And now I see with eye serene
The very pulse of the machine.”*

William Wordsworth
She Was a Phantom of Delight
1807

Machine language: To the uninitiated, mastery of this mysterious tongue is the capstone of computer guruhood. To us machine language programmers (ahem!), it doesn't seem all that difficult.

Thirty of this book's BASIC programs include machine language routines POKEd into memory from DATA statements. Those programs let you see the power of machine language, but their cryptic DATA statements totally conceal its structure.

Well, now that structure is revealed: this chapter includes machine language disassembly listings for every one of the 30 programs!

There are also maps of the memory locations they use, plus a group of Machine Language Monitor tricks and a few simple programming tips.



Printouts from Machine Language Monitors

The 128 and Plus/4's built-in Machine Language Monitors are wonderful tools for exploring the computer's memory, as are the many monitors available for the 64.

It's frequently desirable to get a printout of the monitor display, but unfortunately most of the manuals neglect to tell you how to do it. In most cases, it's very easy to get such a printout. Just try the following steps:

- 1) In direct mode, enter OPEN 4,4:CMD4
- 2) Enter your Machine Language Monitor. (On the 128 and Plus/4, this is done by typing MONITOR and pressing Return).
- 3) Type in the monitor command whose response you want printed out. If all is well, what normally would be displayed on your screen will now be sent to your printer.

To discontinue printing, just enter X and press Return. You'll instantly go back to BASIC.

To keep funny things from happening later on, go back to BASIC's direct mode and enter PRINT#4:CLOSE4.

*Richard Murphy
Talequah, Oklahoma*



Machine Language Monitors as Assemblers

The 128 and Plus/4's built-in Machine Language Monitors include a very simple assembler, as do most of the monitors for the 64 and other computers. The assembler lets you type the mnemonic for an instruction, then it converts that mnemonic to the numbers that are actually poked into memory.

While every "real" assembler makes it easy to insert instructions between already existing ones, these simple implementations lack that very important feature. So when you use them to change a program, you usually have to retype every byte that follows the change. If you make many changes at all, all those retypings can get tiresome pretty quickly.

Well, I use the monitor's T command to get away from retyping. This command copies a block of memory to any other location, leaving the original block unchanged. So when I write a piece of machine code, I always T it to another area in memory. Once it's there, I can change the original code to my heart's content; if the changes consist of insertions, I type them into the original code, then "re-T" the remainder

from its temporary location back to the end of the insertions. I may have to clean up some relative branches, but that's usually a lot easier than retyping all the rest of the program.

In addition to easing insertions, the T procedure lets me easily restore the program to its original state, just by "re-T-ing" the entire block of code. That's useful when my changes are for the worse rather than the better, which happens more often than I like.

*Robert J. Tiess
Middletown, New York*

○☆ **128 Machine Language Monitor and Resets**

It doesn't say so in the manual, but when resetting the 128 you can go directly to the Monitor simply by holding down the Run/Stop key as you press the Reset button. (This also works on the Plus/4).

You can use this if your BASIC program crashes in a seemingly unrecoverable way. To recover harmlessly from the crash, depress Run/Stop and push the Reset button. When the Monitor prompt appears, type an X and press Return. The machine will return to BASIC, with your program still intact.

*Anna Mae Hertzler
Boynton Beach, Florida*

☆ **128 Machine Language Monitor for 64 Mode**

The 128's Machine Language Monitor can also be used on the 64 side of the computer. The way I do it is to write the 64 machine language program while in 128 mode, observing all the rules and memory locations for the 64. When I'm finished, I go to 64 mode by hitting the Reset button while holding the Commodore key. I then SYS to the start of the machine language program and see the results of my handiwork.

*Joe Grossinger
El Paso, Texas*

△☆ **Supermon Revealed**

The public domain Supermon program is one of the most familiar Machine Language Monitors for the 64. Most user groups and bulletin board systems have at least one version available for their members.

If you know Supermon's commands, you can use it to work with machine language (ML) programs. But if you also know something about

its internal workings, your Supermon experience can be eminently more rewarding, especially after a program crash. Read on and learn the details.

When the BASIC version of Supermon is run, it checks the top-of-BASIC memory pointer and creates an ML monitor program below that address. It then lowers the pointer to protect the ML, points the BRK vector to the ML's entry point, and executes a BRK. That displays the B* and register display, plus the dot prompt for the next Supermon command.

After that happens, Supermon's original BASIC program is no longer needed, so you can NEW it or write over it at will.

The above-mentioned BRK vector is in addresses 790-791. Any time the computer executes a machine language BRK, it looks at the address in the BRK vector, then jumps there. In this case, that jump will activate Supermon. From BASIC, a simple way to execute an ML BRK is to SYS to a location containing a zero. SYS8 is easy to remember and to type.

If you crash when using ML, pressing Stop/Restore or a Reset button will return the BRK vector to its power-up value. Supermon's ML will still be at the top of memory, but a BRK will no longer activate it. You can fix everything up with a SYS to the top-of-BASIC pointer, as reset by Supermon. Before a crash, you can find it by:

```
PRINT PEEK(55) + 256*PEEK(56)
```

After recovering from a crash, just SYS to that number, and Supermon will be as good as new.

If you have a Reset button, using it will return the top-of-BASIC pointer to its power-up value, which means that Supermon's ML is no longer in a protected area of memory. If you want to restore the protection, record Supermon's pre-crash values for locations 55 and 56, then restore them after using your Reset button.

*Frank Biondi
Mount Lebanon, Pennsylvania*



Machine Language String Printer

If you program in machine language on the 64, here's a short routine that will print out a string. The string must be in memory starting at location START, and it must end in a zero byte (\$00).

```
LDA# lo byte of START  
LDX# hi byte of START  
JSR $ABLE
```

David Grabowski
Westfield, New Jersey



Machine Language Screen Printing

Here's a little trick for machine language users. Instead of using JSR \$FFD2 to print ASCII characters on the screen, you can use JSR \$E716 to output to the screen regardless of the current output device. This way you can do things like sending messages to the screen while outputting to the printer.

Gregory Houston
Clanton, Alabama



Memory Maps

The machine language programs in this chapter use many special memory locations in the computer. Tables 20-1 and 20-2 are lists of those locations, along with brief descriptions of their usual purpose.

More thorough descriptions and detailed memory maps for the 64 and 128 can be found in the Programmer's Reference Guides and other specialized books.

Table 20-1. 64 Memory locations used by programs in this chapter.

ZERO-PAGE RAM MEMORY LOCATIONS:

\$0001	ON-CHIP 8-BIT I/O REGISTER
\$0002	UNUSED
\$000A	FLAG: 0=LOAD, 1=VERIFY
\$000F	FLAG: DATA SCAN / LIST QUOTE / GARBAGE COLLECTION
\$0014-\$0015	TEMP: INTEGER VALUE
\$0022	UTILITY POINTER
\$002B-\$002C	POINTER: START OF BASIC TEXT
\$002D-\$002E	POINTER: START OF BASIC VARIABLES
\$002F-\$0030	POINTER: START OF BASIC ARRAYS
\$0037-\$0038	POINTER: HIGHEST ADDRESS USED BY BASIC
\$0049-\$004A	POINTER: INDEX VARIABLE FOR FOR...NEXT
\$004B-\$0060	TEMPORARY POINTER/DATA AREA
\$0062-\$0065	FLOATING POINT ACCUMULATOR #1: MANTISSA
\$0066	FLOATING POINT ACCUMULATOR #1: SIGN
\$0091	FLAG: STOP KEY / RVS KEY
\$009A	DEFAULT OUTPUT DEVICE
\$00BA	CURRENT DEVICE NUMBER
\$00C5	CURRENT KEY PRESSED. 0=NO KEY
\$00C6	NUMBER OF CHARACTERS IN KEYBOARD BUFFER

Chapter 20

Table 20-1. Continued.

\$00CB	FLAG: PRINT SHIFTED CHARACTERS
\$00D7	TEMPORARY DATA AREA / LAST KEY PRESSED
\$00D8	FLAG: INSERT MODE (>0 = # OF INSERTS)
\$00F9-\$00FA	RS232 OUTPUT BUFFER POINTER
\$00FB-\$00FE	FREE SPACE FOR USER PROGRAMS

OTHER RAM MEMORY LOCATIONS:

\$0277-\$0280	KEYBOARD BUFFER
\$0286	CURRENT CHARACTER COLOR
\$028D	FLAG: SHIFT / CTRL / COMMODORE KEYS
\$02A7-\$02FF	UNUSED
\$0306-\$0307	VECTOR: BASIC TEXT LIST
\$0314-\$0315	VECTOR: HARDWARE IRQ INTERRUPT
\$0330-\$0331	KERNAL LOAD ROUTINE VECTOR
\$0332-\$0333	KERNAL SAVE ROUTINE VECTOR
\$0334-\$03FF	TAPE BUFFER / UNUSED
\$0400-\$07FF	SCREEN MEMORY

BASIC ROM:

\$A437	GENERAL ERROR HANDLER
\$A613	SEARCH FOR LINE NUMBER
\$A642	PERFORM NEW
\$A659	PART OF LIST ROUTINE
\$A71A	PART OF TOKEN PRINT ROUTINE
\$AAD7	PART OF PRINT ROUTINE
\$AB1E	PRINT STRING (USES Y AND A REGISTERS)
\$AB47	PART OF PRINT STRING ROUTINE
\$AD9E	EVALUATE EXPRESSION
\$AEFD	PART OF CHECK PARENTHESES ROUTINE
\$B391	CONVERT INTEGER TO FLOATING POINT
\$B6A6	PART OF STRING DISCARD ROUTINE
\$B79B	INPUT A PARAMETER
\$BDCD	OUTPUT A NUMBER IN DECIMAL DIGITS
\$BDDD	CONVERT CONTENTS OF FAC1 TO A STRING

USER RAM:

\$C000-\$CFFF	UNUSED RAM AREA
---------------	-----------------

VIC, SID, I/O DEVICES AND COLOR RAM:

\$D011	VERTICAL FINE SCROLLING AND CONTROL REGISTER
--------	--

Table 20-1. Continued.

\$D012	READ/WRITE RASTER VALUE FOR COMPARE IRQ
\$D018	VIC MEMORY CONTROL REGISTER
\$D019	VIC INTERRUPT FLAG REGISTER
\$D020	BORDER COLOR REGISTER
\$D021	BACKGROUND COLOR 0 REGISTER
\$D401	VOICE 1 FREQUENCY CONTROL (HIGH BYTE)
\$D404	VOICE 1 CONTROL REGISTER
\$D418	VOLUME AND FILTER SELECT REGISTER
\$D800-\$DBFF	COLOR RAM
\$DC00	CIA #1 DATA PORT A
\$DC01	CIA #1 DATA PORT B
\$DC08	TOD CLOCK TENTHS OF SECONDS
\$DC09	TOD CLOCK SECONDS
\$DC0A	TOD CLOCK MINUTES
\$DC0B	TOD CLOCK HOURS
\$DC0E	CONTROL REGISTER A

KERNAL ROM:

\$E0F9	CALL KERNAL I/O ROUTINES
\$E1D4	SET PARAMETERS FOR LOAD, VERIFY AND SAVE
\$E716	OUTPUT TO THE SCREEN
\$EA31	IRQ INTERRUPT ENTRY
\$F4A5	PRINT "SEARCHING" MESSAGE IF IN DIRECT MODE
\$F5ED	PART OF SAVE RAM ROUTINE
\$FCE2	POWER-UP RESET ROUTINE

KERNAL JUMP TABLE:

\$FF81	INITIALIZE SCREEN EDITOR AND VIC CHIP
\$FF84	INITIALIZE I/O DEVICES
\$FF8A	RESTORE DEFAULT I/O VECTORS
\$FF90	SET KERNAL MESSAGE CONTROL FLAG
\$FFB7	READ I/O STATUS WORD
\$FFBA	SET LOGICAL FILE PARAMETERS
\$FFBD	SET FILENAME PARAMETERS
\$FFC0	OPEN A LOGICAL FILE
\$FFC6	DEFINE AN INPUT CHANNEL
\$FFCC	RESTORE DEFAULT DEVICES
\$FFCF	INPUT A CHARACTER
\$FFD2	OUTPUT A CHARACTER
\$FFD5	LOAD FROM A DEVICE
\$FFE4	GET A CHARACTER
\$FFE7	CLOSE ALL FILES
\$FFF0	READ/SET CURSOR POSITION

Table 20-2. 128 Memory locations used by programs in this chapter.

\$002D-\$002E	POINTER: START OF BASIC TEXT
\$00D8	TEXT/GRAPHIC MODE FLAG
\$00F1	DEFAULT FOREGROUND COLOR
\$0314-\$0315	IRQ RAM VECTOR
\$0B00-\$0BFF	TAPE BUFFER/PROGRAM STORAGE AREA
\$1210-\$1211	POINTER: TOP OF TEXT
\$4112	INITIAL
\$417A	INITIALIZE D501-4
\$418D	PART OF BASIC ROM
\$4D37	PART OF BASIC ROM
\$5238	PART OF BASIC ROM
\$D011	VIC CONTROL REGISTER 1
\$D012	READ/WRITE RASTER VALUE FOR COMPARE IRQ
\$D019	VIC INTERRUPT FLAG REGISTER
\$D021	BACKGROUND COLOR 1
\$FA65	IRQ
\$FF81	INIT EDITOR & DISPLAY CHIPS
\$FF84	INIT I/O DEVICES
\$FF8A	RESTORE VECTORS TO INITIAL SYSTEM

○△☆ Digging into Machine Language

The main obstacle to understanding machine language is its foreignness, especially when compared to BASIC. It doesn't use English words, and it doesn't use familiar concepts of mathematics. And the so-called object code is nothing more than a series of hexadecimal numbers. It's almost un-American. But if you overcome those minor psychological barriers, machine language is simple, straightforward and beautiful. The way to overcome them is to dig into some simple programs.

Figures 20-1 through 20-30 are disassembly listings of the machine language generated by the BASIC programs in the other chapters of this book. They appear alphanumerically by the name of the BASIC program.

Each line of a disassembly contains one machine language instruction, composed of the following parts:

- The hexadecimal address of the first byte of the instruction (in Figure 20-1's 64 Appender, this is 02BC).
- One, two, or three hexadecimal numerals representing the instruction itself (the two bytes A0 00 in 64 Appender).
- The three-letter assembler mnemonic for the machine language instruction itself (LDY in our example, for Load Y).

- The *operand*, or number on which the instruction is to operate (hex \$00 in our example. The “#” denotes an immediate instruction, or one in which the quantity 00 is operated on, rather than the contents of memory location 00).

While disassembly listings can show you a program's structure, there are even better tools for understanding it. Figure 20-23's listing for 64 SIXPACK PLUS is an example of a completely commented source listing; notice how much more information it contains.

Source listings are gotten from the creator of the program (if he or she will release them), while disassembly listings are made by converting the machine language into its assembly language equivalent.

If you know even a little machine language, you should be able to follow the logic in most of these disassemblies. Following that logic leads to the ability to analyze it, which in turn leads to the ability to modify the program to fit your own desires. Happy digging!

64 APPENDER (page 339)

	02BC	A0 00	LDY #\$00
	02BE	E6 2B	INC \$2B
	02C0	D0 02	BNE \$02C4
	02C2	E6 2C	INC \$2C
	02C4	B1 2B	LDA (\$2B),Y
	02C6	D0 F6	BNE \$02BE
	02C8	E6 2B	INC \$2B
Fig. 20-1.	02CA	D0 02	BNE \$02CE
	02CC	E6 2C	INC \$2C
	02CE	B1 2B	LDA (\$2B),Y
	02D0	D0 EC	BNE \$02BE
	02D2	60	RTS
	02D3	A9 01	LDA #\$01
	02D5	85 2B	STA \$2B
	02D7	A9 08	LDA #\$08
	02D9	85 2C	STA \$2C
	02DB	60	RTS

64 BASIC LINKER (page 336)

	C000	20	FD	AE	JSR	\$AEFD
	C003	A9	00		LDA	#\$00
	C005	85	0A		STA	\$0A
	C007	20	D4	E1	JSR	\$E1D4
	C00A	A5	0A		LDA	\$0A
	C00C	A6	2B		LDX	\$2B
	C00E	A4	2C		LDY	\$2C
	C010	20	D5	FF	JSR	\$FFD5
	C013	B0	21		BCS	\$C036
	C015	20	42	A6	JSR	\$A642
	C018	20	B7	FF	JSR	\$FFB7
Fig. 20-2.	C01B	29	BF		AND	#\$BF
	C01D	F0	05		BEQ	\$C024
	C01F	A2	1D		LDX	#\$1D
	C021	4C	37	A4	JMP	\$A437
	C024	86	2D		STX	\$2D
	C026	86	2F		STX	\$2F
	C028	84	2E		STY	\$2E
	C02A	84	30		STY	\$30
	C02C	A9	00		LDA	#\$00
	C02E	20	90	FF	JSR	\$FF90
	C031	68			PLA	
	C032	68			PLA	
	C033	4C	59	A6	JMP	\$A659
	C036	4C	F9	E0	JMP	\$E0F9

64 BINARY KEY DECODER (page 256)

	C000	20	E4	FF	JSR	\$FFE4
	C003	F0	FB		BEQ	\$C000
	C005	20	D2	FF	JSR	\$FFD2
	C008	85	FB		STA	\$FB
	C00A	A9	11		LDA	#\$11
	C00C	20	D2	FF	JSR	\$FFD2
	C00F	A5	FB		LDA	\$FB
Fig. 20-3.	C011	85	FB		STA	\$FB
	C013	A2	08		LDX	#\$08
	C015	06	FB		ASL	\$FB
	C017	A9	30		LDA	#\$30
	C019	69	00		ADC	#\$00
	C01B	20	D2	FF	JSR	\$FFD2
	C01E	CA			DEX	
	C01F	D0	F4		BNE	\$C015
	C021	60			RTS	

64 BORDER FIREWORKS (page 225)

Fig. 20-4.

02A7	AC	20	D0	LDY	\$D020
02AA	CA			DEX	
02AB	8E	20	D0	STX	\$D020
02AE	A5	C6		LDA	\$C6
02B0	F0	F8		BEQ	\$02AA
02B2	8C	20	D0	STY	\$D020
02B5	60			RTS	

64 COLOR FLIPPER (page 221)

Fig. 20-5.

CF39	AD	5E	CF	LDA	\$CF5E
CF3C	4D	5F	CF	EOR	\$CF5F
CF3F	8D	60	CF	STA	\$CF60
CF42	A9	00		LDA	#\$00
CF44	85	FB		STA	\$FB
CF46	A9	D8		LDA	#\$D8
CF48	85	FC		STA	\$FC
CF4A	A2	04		LDX	#\$04
CF4C	A0	00		LDY	#\$00
CF4E	B1	FB		LDA	(\$FB),Y
CF50	4D	60	CF	EOR	\$CF60
CF53	91	FB		STA	(\$FB),Y
CF55	C8			INY	
CF56	D0	F6		BNE	\$CF4E
CF58	E6	FC		INC	\$FC
CF5A	CA			DEX	
CF5B	D0	EF		BNE	\$CF4C
CF5D	60			RTS	

64 COLOR SYS (page 223)

Fig. 20-6.

033C	20	9B	B7	JSR	\$B79B
033F	8A			TXA	
0340	48			PHA	
0341	20	9B	B7	JSR	\$B79B
0344	8A			TXA	
0345	48			PHA	
0346	20	9B	B7	JSR	\$B79B
0349	8A			TXA	
034A	8D	86	02	STA	\$0286
034D	68			PLA	
034E	8D	21	D0	STA	\$D021
0351	68			PLA	
0352	8D	20	D0	STA	\$D020
0355	60			RTS	
0356	00			BRK	

64 DELETE RIGHT (page 40)

	02A8	78		SEI
	02A9	A2	B5	LDX #B5
	02AB	A0	02	LDY #02
	02AD	8E	14 03	STX \$0314
	02B0	8C	15 03	STY \$0315
	02B3	58		CLI
	02B4	60		RTS
	02B5	48		PHA
	02B6	8A		TXA
	02B7	48		PHA
	02B8	98		TYA
	02B9	48		PHA
Fig. 20-7.	02BA	A5	C5	LDA \$C5
	02BC	C9	03	CMP #03
	02BE	D0	0E	BNE \$02CE
	02C0	A9	02	LDA #02
	02C2	85	C6	STA \$C6
	02C4	A9	1D	LDA #1D
	02C6	8D	77 02	STA \$0277.
	02C9	A9	14	LDA #14
	02CB	8D	78 02	STA \$0278
	02CE	68		PLA
	02CF	A8		TAY
	02D0	68		PLA
	02D1	AA		TAX
	02D2	68		PLA
	02D3	4C	31 EA	JMP \$EA31

64 DISK DEFAULTER (page 320)

	02A7	A9	02	LDA #02
	02A9	8D	31 03	STA \$0331
	02AC	8D	33 03	STA \$0333
	02AF	A9	BA	LDA #BA
	02B1	8D	30 03	STA \$0330
	02B4	A9	C5	LDA #C5
	02B6	8D	32 03	STA \$0332
Fig. 20-8.	02B9	60		RTS
	02BA	A9	08	LDA #08
	02BC	85	BA	STA \$BA
	02BE	A9	00	LDA #00
	02C0	85	0A	STA \$0A
	02C2	4C	A5 F4	JMP \$F4A5
	02C5	A9	08	LDA #08
	02C7	85	BA	STA \$BA
	02C9	4C	ED F5	JMP \$F5ED
	02CC	0A		ASL

64 DISK ERROR READER (page 321)

Fig. 20-9.

```

033E A9 0F LDA #0F
0340 A8 TAY
0341 A2 08 LDX #08
0343 20 BA FF JSR $FFBA
0346 A9 00 LDA #00
0348 20 BD FF JSR $FFBD
034B 20 C0 FF JSR $FFC0
034E A2 0F LDX #0F
0350 20 C6 FF JSR $FFC6
0353 20 CF FF JSR $FFCF
0356 C9 0D CMP #0D
0358 F0 05 BEQ $035F
035A 20 D2 FF JSR $FFD2
035D D0 F4 BNE $0353
035F 20 E7 FF JSR $FFE7
0362 4C CC FF JMP $FFCC

```

64 EASY WINDOWS (page 223)

Fig. 20-10.

```

033C 20 9B B7 JSR $B79B
033F 8A TXA
0340 48 PHA
0341 20 9B B7 JSR $B79B
0344 8A TXA
0345 48 PHA
0346 20 9B B7 JSR $B79B
0349 8A TXA
034A 8D 86 02 STA $0286
034D 68 PLA
034E 8D 21 D0 STA $D021
0351 68 PLA
0352 8D 20 D0 STA $D020
0355 60 RTS
0356 00 BRK
0357 CD 99 03 CMP $0399
035A F0 2D BEQ $0389
035C EE 86 02 INC $0286
035F AD 86 02 LDA $0286
0362 A0 00 LDY #00
0364 99 00 D8 STA $D800,Y
0367 99 00 D9 STA $D900,Y
036A 99 00 DA STA $DA00,Y
036D 99 00 DB STA $DB00,Y
0370 C8 INY
0371 F0 16 BEQ $0389
0373 4C 64 03 JMP $0364
0376 CD 98 03 CMP $0398
033C A0 00 LDY #00

```

```

033E B9 00 04 LDA $0400,Y
0341 99 00 C0 STA $C000,Y
0344 B9 00 05 LDA $0500,Y
0347 99 00 C1 STA $C100,Y
034A B9 00 06 LDA $0600,Y
034D 99 00 C2 STA $C200,Y
0350 B9 E8 06 LDA $06E8,Y
0353 99 E8 C2 STA $C2E8,Y
0356 C8      INY
0357 D0 E5    BNE $033E
0359 60      RTS
035A A0 00     LDY #$00
035C B9 00 C0 LDA $C000,Y
035F 99 00 04 STA $0400,Y
0362 B9 00 C1 LDA $C100,Y
0365 99 00 05 STA $0500,Y
0368 B9 00 C2 LDA $C200,Y
036B 99 00 06 STA $0600,Y
036E B9 E8 C2 LDA $C2E8,Y
0371 99 E8 06 STA $06E8,Y
0374 C8      INY
0375 D0 E5    BNE $035C
0377 60      RTS

```

Fig. 20-10. Continued.

```

64 F7 APOSTROPHE (page 41)
C38C 78      SEI
C38D A2 99     LDX #$99
C38F A0 C3     LDY #$C3
C391 8E 14 03 STX $0314
C394 8C 15 03 STY $0315
C397 58      CLI
C398 60      RTS
C399 48      PHA
C39A 8A      TXA
C39B 48      PHA
C39C 98      TYA
C39D 48      PHA
C39E A5 D7     LDA $D7
C3A0 C9 88     CMP #$88
C3A2 D0 05     BNE $C3A9
C3A4 A9 27     LDA #$27
C3A6 20 D2 FF JSR $FFD2
C3A9 68      PLA
C3AA A8      TAY
C3AB 68      PLA
C3AC AA      TAX
C3AD 68      PLA
C3AE 4C 31 EA JMP $EA31

```

Fig. 20-11.

64 F-KEY COLORS (page 222)

033C	78		SEI
033D	A9	49	LDA #S49
033F	8D	14 03	STA S0314
0342	A9	03	LDA #S03
0344	8D	15 03	STA S0315
0347	58		CLI
0348	60		RTS
0349	A5	C5	LDA SC5
034B	C9	04	CMP #S04
034D	F0	32	BEQ S0381
034F	C9	05	CMP #S05
0351	F0	23	BEQ S0376
0353	C9	06	CMP #S06
0355	D0	32	BNE S0389
0357	CD	99 03	CMP S0399
035A	F0	2D	BEQ S0389
035C	EE	86 02	INC S0286
035F	AD	86 02	LDA S0286
0362	A0	00	LDY #S00
0364	99	00 D8	STA SD800, Y
0367	99	00 D9	STA SD900, Y
036A	99	00 DA	STA SDA00, Y
036D	99	00 DB	STA SDB00, Y
0370	C8		INY
0371	F0	16	BEQ S0389
0373	4C	64 03	JMP S0364
0376	CD	98 03	CMP S0398
0379	F0	0E	BEQ S0389
037B	EE	21 D0	INC SD021
037E	4C	89 03	JMP S0389
0381	CD	97 03	CMP S0397
0384	F0	03	BEQ S0389
0386	EE	20 D0	INC SD020
0389	A5	C5	LDA SC5
038B	8D	97 03	STA S0397
038E	8D	98 03	STA S0398
0391	8D	99 03	STA S0399
0394	4C	31 EA	JMP SEA31
0397	00		BRK
0398	00		BRK
0399	00		BRK
039A	00		BRK

Fig. 20-12.

64 KEYBEEPER (page 243)

Fig. 20-13.

C350	78			SEI
C351	A9	62		LDA # \$62
C353	8D	14	03	STA \$0314
C356	A9	C3		LDA # \$C3
C358	8D	15	03	STA \$0315
C35B	58			CLI
C35C	A9	0F		LDA # \$0F
C35E	8D	18	D4	STA \$D418
C361	60			RTS
C362	A5	C5		LDA \$C5
C364	A4	FE		LDY \$FE
C366	88			DEY
C367	C9	40		CMP # \$40
C369	F0	16		BEQ \$C381
C36B	A2	80		LDX # \$80
C36D	C9	01		CMP # \$01
C36F	F0	0C		BEQ \$C37D
C371	A2	40		LDX # \$40
C373	C9	08		CMP # \$08
C375	90	06		BCC \$C37D
C377	C9	33		CMP # \$33
C379	F0	02		BEQ \$C37D
C37B	A2	10		LDX # \$10
C37D	8E	01	D4	STX \$D401
C380	C8			INY
C381	8C	04	D4	STY \$D404
C384	4C	31	EA	JMP \$EA31

64 MACHINE LANGUAGE LOADER (page 338)

Fig. 20-14.

0334	20	FD	AE	JSR \$AEFD
0337	A9	01		LDA # \$01
0339	A2	08		LDX # \$08
033B	A8			TAY
033C	20	BA	FF	JSR \$FFBA
033F	20	9E	AD	JSR \$AD9E
0342	A9	0D		LDA # \$0D
0344	D0	05		BNE \$034B
0346	A2	16		LDX # \$16
0348	4C	37	A4	JMP \$A437
034B	20	A6	B6	JSR \$B6A6
034E	20	BD	FF	JSR \$FFBD
0351	A9	00		LDA # \$00
0353	20	D5	FF	JSR \$FFD5
0356	60			RTS

64 MIRROR (page 288)

033C	AD 0E DC	LDA \$DC0E
033F	29 FE	AND #\$FE
0341	8D 0E DC	STA \$DC0E
0344	A5 01	LDA \$01
0346	29 FB	AND #\$FB
0348	85 01	STA \$01
034A	A9 00	LDA #\$00
034C	85 FB	STA \$FB
034E	85 FD	STA \$FD
0350	A9 D0	LDA #\$D0
0352	85 FC	STA \$FC
0354	A9 30	LDA #\$30
0356	85 FE	STA \$FE
0358	A2 08	LDX #\$08
035A	86 FA	STX \$FA
035C	A0 00	LDY #\$00
035E	B1 FB	LDA (\$FB), Y
0360	A2 08	LDX #\$08
0362	6A	ROR
0363	26 02	ROL \$02
0365	CA	DEX
0366	D0 FA	BNE \$0362
0368	A5 02	LDA \$02
036A	91 FD	STA (\$FD), Y
036C	C8	INY
036D	D0 EF	BNE \$035E
036F	E6 FC	INC \$FC
0371	E6 FE	INC \$FE
0373	C6 FA	DEC \$FA
0375	D0 E7	BNE \$035E
0377	A5 01	LDA \$01
0379	09 04	ORA #\$04
037B	85 01	STA \$01
037D	AD 0E DC	LDA \$DC0E
0380	09 01	ORA #\$01
0382	8D 0E DC	STA \$DC0E
0385	AD 18 D0	LDA \$D018
0388	29 F0	AND #\$F0
038A	09 0C	ORA #\$0C
038C	8D 18 D0	STA \$D018
038F	A9 30	LDA #\$30
0391	85 38	STA \$38
0393	60	RTS

Fig. 20-15.

64 PAUSER (page 333)

	02A7	AD	14	03	LDA	\$0314
	02AA	8D	CA	02	STA	\$02CA
	02AD	AD	15	03	LDA	\$0315
	02B0	8D	CB	02	STA	\$02CB
	02B3	78			SEI	
	02B4	A9	02		LDA	#\$02
	02B6	8D	15	03	STA	\$0315
	02B9	A9	C0		LDA	#\$C0
	02BB	8D	14	03	STA	\$0314
	02BE	58			CLI	
	02BF	60			RTS	
	02C0	AD	8D	02	LDA	\$028D
Fig. 20-16.	02C3	29	05		AND	#\$05
	02C5	C9	05		CMP	#\$05
	02C7	F0	03		BEQ	\$02CC
	02C9	4C	31	EA	JMP	EA31
	02CC	EE	20	D0	INC	\$D020
	02CF	A9	FD		LDA	\$\$FD
	02D1	8D	00	DC	STA	\$DC00
	02D4	AD	01	DC	LDA	\$DC01
	02D7	29	80		AND	#\$80
	02D9	F0	F4		BEQ	\$02CF
	02DB	A9	FF		LDA	\$\$FF
	02DD	8D	00	DC	STA	\$DC00
	02E0	CE	20	D0	DEC	\$D020
	02E3	A9	00		LDA	#\$00
	02E5	8D	8D	02	STA	\$028D
	02E8	4C	C9	02	JMP	\$02C9

64 PRINT@ (page 224)

	0352	20	9B	B7	JSR	\$B79B
	0355	8A			TXA	
	0356	48			PHA	
Fig. 20-17.	0357	20	9B	B7	JSR	\$B79B
	035A	68			PLA	
	035B	AA			TAX	
	035C	A4	65		LDY	\$65
	035E	18			CLC	
	035F	4C	F0	FF	JMP	FFFF0

64 QUICK ERASE (page 334)

Fig. 20-18.

033C	A9 00	LDA #00
033E	85 FB	STA \$FB
0340	A9 08	LDA #08
0342	85 FC	STA \$FC
0344	A9 00	LDA #00
0346	A8	TAY
0347	91 FB	STA (\$FB),Y
0349	C8	INY
034A	C4 FB	CPY \$FB
034C	D0 F9	BNE \$0347
034E	E6 FC	INC \$FC
0350	A9 FA	LDA #\$FA
0352	C5 FC	CMP \$FC
0354	D0 EE	BNE \$0344
0356	20 E2 FC	JSR \$FCE2
0359	60	RTS

64 REMVERT (page 333)

Fig. 20-19.

CF08	A5 2B	LDA \$2B
CF0A	85 FB	STA \$FB
CF0C	A5 2C	LDA \$2C
CF0E	85 FC	STA \$FC
CF10	A0 03	LDY #03
CF12	C8	INY
CF13	B1 FB	LDA (\$FB),Y
CF15	F0 1F	BEQ \$CF36
CF17	C9 8F	CMP #\$8F
CF19	F0 02	BEQ \$CF1D
CF1B	D0 F5	BNE \$CF12
CF1D	C8	INY
CF1E	B1 FB	LDA (\$FB),Y
CF20	F0 14	BEQ \$CF36
CF22	C9 20	CMP #\$20
CF24	D0 0C	BNE \$CF32
CF26	C8	INY
CF27	B1 FB	LDA (\$FB),Y
CF29	C9 20	CMP #\$20
CF2B	F0 F9	BEQ \$CF26
CF2D	C9 12	CMP #\$12
CF2F	F0 01	BEQ \$CF32
CF31	88	DEY
CF32	A9 12	LDA #\$12
CF34	91 FB	STA (\$FB),Y
CF36	A0 01	LDY #01
CF38	B1 FB	LDA (\$FB),Y
CF3A	AA	TAX

Fig. 20-19. Continued.

CF3B	88		DEY
CF3C	B1	FB	LDA (\$FB),Y
CF3E	85	FB	STA \$FB
CF40	86	FC	STX \$FC
CF42	E0	00	CPX #\$00
CF44	D0	CA	BNE \$CF10
CF46	60		RTS

64 SCREEN BARS (page 221)

Fig. 20-20.

C000	A9	93		LDA	#\$93
C002	20	4A	C0	JSR	\$C04A
C005	A2	00		LDX	#\$00
C007	A0	00		LDY	#\$00
C009	A9	00		LDA	#\$00
C00B	8D	20	D0	STA	\$D020
C00E	8D	21	D0	STA	\$D021
C011	EE	0A	C0	INC	\$C00A
C014	AD	0A	C0	LDA	\$C00A
C017	C9	11		CMP	#\$11
C019	D0	05		BNE	\$C020
C01B	A9	00		LDA	#\$00
C01D	8D	0A	C0	STA	\$C00A
C020	20	33	C0	JSR	\$C033
C023	E8			INX	
C024	E0	FF		CPX	#\$FF
C026	D0	E1		BNE	\$C009
C028	EE	60	C0	INC	\$C060
C02B	AD	60	C0	LDA	\$C060
C02E	C9	02		CMP	#\$02
C030	D0	D7		BNE	\$C009
C032	60			RTS	
C033	C8			INY	
C034	C0	FF		CPY	#\$FF
C036	D0	FB		BNE	\$C033
C038	EE	61	C0	INC	\$C061
C03B	AD	61	C0	LDA	\$C061
C03E	C9	02		CMP	#\$02
C040	D0	F1		BNE	\$C033
C042	A9	00		LDA	#\$00
C044	8D	61	C0	STA	\$C061
C047	60			RTS	
C048	EA			NOP	
C049	EA			NOP	
C04A	20	D2	FF	JSR	\$\$FFD2
C04D	A9	00		LDA	#\$00
C04F	8D	60	C0	STA	\$C060
C052	60			RTS	

64 SCREEN EFFECT (page 225)

Fig. 20-21.

C000	C8		INY
C001	8C	21 D0	STY \$D021
C004	8C	20 D0	STY \$D020
C007	A2	5E	LDX #\$5E
C009	CA		DEX
C00A	10	FD	BPL \$C009
C00C	A5	CB	LDA \$CB
C00E	C9	3F	CMP #\$3F
C010	D0	EE	BNE \$C000
C012	A5	CB	LDA \$CB
C014	C9	40	CMP #\$40
C016	D0	FA	BNE \$C012
C018	60		RTS

64 SEARCH (page 335)

Fig. 20-22.

C000	A0	00	LDY #\$00
C002	C8		INY
C003	B1	7A	LDA (\$7A),Y
C005	99	CD C0	STA \$C0CD,Y
C008	D0	F8	BNE \$C002
C00A	A0	00	LDY #\$00
C00C	98		TYA
C00D	99	00 02	STA \$0200,Y
C010	C8		INY
C011	C0	50	CPY #\$50
C013	D0	F8	BNE \$C00D
C015	A9	08	LDA #\$08
C017	85	FC	STA \$FC
C019	A9	01	LDA #\$01
C01B	85	FB	STA \$FB
C01D	A0	02	LDY #\$02
C01F	B1	FB	LDA (\$FB),Y
C021	8D	CB C0	STA \$C0CB
C024	C8		INY
C025	B1	FB	LDA (\$FB),Y
C027	8D	CC C0	STA \$C0CC
C02A	C8		INY
C02B	B1	FB	LDA (\$FB),Y
C02D	F0	1B	BEQ \$C04A
C02F	CD	CE C0	CMP \$C0CE
C032	D0	F6	BNE \$C02A
C034	8C	CD C0	STY \$C0CD
C037	A2	00	LDX #\$00

C039	E8			INX
C03A	C8			INY
C03B	BD	CE	C0	LDA \$C0CE,X
C03E	F0	1C		BEQ \$C05C
C040	D1	FB		CMP (\$FB),Y
C042	F0	F5		BEQ \$C039
C044	AC	CD	C0	LDY \$C0CD
C047	4C	2A	C0	JMP \$C02A
C04A	C8			INY
C04B	18			CLC
C04C	98			TYA
C04D	65	FB		ADC \$FB
C04F	85	FB		STA \$FB
C051	90	02		BCC \$C055
C053	E6	FC		INC \$FC
C055	A0	01		LDY #\$01
C057	B1	FB		LDA (\$FB),Y
C059	D0	C2		BNE \$C01D
C05B	60			RTS
C05C	AD	CB	C0	LDA \$C0CB
C05F	85	14		STA \$14
C061	AD	CC	C0	LDA \$C0CC
C064	85	15		STA \$15
C066	20	13	A6	JSR \$A613
C069	A0	01		LDY #\$01
C06B	84	0F		STY \$0F
C06D	20	D7	AA	JSR \$AAD7
C070	A0	03		LDY #\$03
C072	84	49		STY \$49
C074	AD	CC	C0	LDA \$C0CC
C077	AE	CB	C0	LDX \$C0CB
C07A	20	CD	BD	JSR \$BD CD
C07D	A9	20		LDA #\$20
C07F	A4	49		LDY \$49
C081	29	7F		AND #\$7F
C083	20	47	AB	JSR \$AB47
C086	C9	22		CMP #\$22
C088	D0	06		BNE \$C090
C08A	A5	0F		LDA \$0F
C08C	49	FF		EOR #\$FF
C08E	85	0F		STA \$0F
C090	C8			INY
C091	F0	04		BEQ \$C097
C093	B1	5F		LDA (\$5F),Y
C095	D0	0B		BNE \$C0A2
C097	A5	60		LDA \$60
C099	85	FC		STA \$FC

Fig. 20-22. Continued.

```

C09B A5 5F      LDA $5F
C09D 85 FB      STA $FB
C09F 4C 4A C0   JMP $C04A
C0A2 10 DF      BPL $C083
C0A4 C9 FF      CMP #$FF
C0A6 F0 DB      BEQ $C083
C0A8 24 0F      BIT $0F
C0AA 30 D7      BMI $C083
C0AC 38         SEC
C0AD E9 7F      SBC #$7F
C0AF AA        TAX
C0B0 84 49      STY $49
C0B2 A0 FF      LDY #$FF
C0B4 CA        DEX
C0B5 F0 08      BEQ $C0BF
C0B7 C8        INY
C0B8 B9 9E A0   LDA $A09E,Y
C0BB 10 FA      BPL $C0B7
C0BD 30 F5      BMI $C0B4
C0BF C8        INY
C0C0 B9 9E A0   LDA $A09E,Y
C0C3 30 BA      BMI $C07F
C0C5 20 47 AB   JSR $AB47
C0C8 D0 F5      BNE $C0BF
C0CA 60        RTS

```

Fig. 20-22. Continued.

64 SIXPACK PLUS (page 326)

```

CB00 48          PHA          ;$CB00-$CB05: to be safe,
CB01 8A          TXA          ;present values of all registers
CB02 48          PHA          ;(A,X,Y, and SR)
CB03 98          TYA          ;are saved by pushing
CB04 48          PHA          ;onto the stack
CB05 08          PHP          ;(page 2 temporary storage buffer)
CB06 20 E4 FF   JSR $FFE4     ;get char from kbd buffer, Kernal GETIN
CB09 C9 BF       CMP #BF     ;is it Commodore-B?
CB0B D0 03       BNE $CB10    ;if no, then branch
CB0D EE 20 D0   INC $D020    ;increment border color value by one
CB10 C9 AE       CMP #AE     ;Commodore-S pressed?
CB12 D0 03       BNE $CB17    ;if no, then branch
CB14 EE 21 D0   INC $D021    ;= POKE53281,(PEEK(53281)+1) AND15
CB17 C9 0D       CMP #$0D     ;RETURN pressed?
CB19 D0 EB       BNE $CB06    ;if no, then branch to repeat loop
CB1B A9 00       LDA #$00     ;$CB1B-$CB1E: clears keyboard buffer
CB1D 85 C6       STA $C6     ;effect same as POKE 198,0 in BASIC
CB1F 28          PLP          ;$CB1F-$CB24: restore the stack
CB20 68          PLA          ;and all registers
CB21 A8          TAY          ;(SR,Y,X, and A)
CB22 68          PHA          ;by pulling values

```

Fig. 20-23.

Chapter 20

```

CB23 AA      TAX
CB24 68      PLA
CB25 A2 3A   LDX #$3A
CB27 A0 CB   LDY #$CB
CB29 8E 06 03 STX $0306
CB2C 8C 07 03 STY $0307
CB2F 78      SEI
CB30 A2 46   LDX #$46
CB32 8E 14 03 STX $0314
CB35 8C 15 03 STY $0315
CB38 58      CLI

CB39 60      RTS
CB3A 48      PHA
CB3B AD 8D 02 LDA $028D
CB3E C9 01   CMP #$01
CB40 F0 F9   BEQ $CB3B
CB42 68      PLA
CB43 4C 1A A7 JMP $A71A

CB46 A5 C5   LDA $C5
CB48 C9 40   CMP #$40
CB4A F0 16   BEQ $CB62
CB4C C9 07   CMP #$07
CB4E F0 15   BEQ $CB65
CB50 A5 D7   LDA $D7
CB52 C9 85   CMP #$85
CB54 F0 20   BEQ $CB76
CB56 C9 86   CMP #$86
CB58 F0 1C   BEQ $CB76
CB5A C9 87   CMP #$87
CB5C F0 18   BEQ $CB76
CB5E C9 88   CMP #$88
CB60 F0 14   BEQ $CB76
CB62 4C 31 EA JMP $EA31
CB65 A6 91   LDX $91
CB67 E0 FB   CPX #$FB
CB69 D0 08   BNE $CB73
CB6B A2 18   LDX $18
CB6D A0 00   LDY #$00
CB6F 18      CLC
CB70 20 F0 FF JSR $FFF0

CB73 4C 31 EA JMP $EA31

CB76 A6 9A   LDX $9A
CB78 E0 03   CPX #$03
CB7A D0 F7   BNE $CB73
CB7C 38      SEC
CB7D E9 85   SBC #$85

```

;previously saved
;off stack -- i.e., last in, first out
;\$CB25-\$CB2E: points vector at 774-775
;(IQPLOT) to detour through
;subroutine starting at \$CB3A
;= BASIC's POKE 774,58:POKE 775,192
;\$CB2F-\$CB38: disable IRQ interrupts/
;point IRQ interrupt vector to detour
;interrupts through subroutine at \$CB46
;reenable IRQ similar to this BASIC line:
;POKE56334,0:POKE788,70:POKE789,192:
; POKE5633 4,1
;Return To Subroutine (return to BASIC)
;Push A register (Accumulator) onto stack
;Load Accumulator with value at loc 653
;is SHIFT being pressed?
;if yes, keep checking (Branch if Equal)
;PuLl A register value from stack
;JuMP to QPLOT routine to scroll LISTing
; on screen
;load value at location 197 into A
;any key pressed yet? (\$40=64=no)
;if no keypress yet, then exit detour
;is CRSR down key pressed?
;if yes, then branch
;load value at location 215 into A
;COMPare with the value for the f1 key?
;Branch if Equal
;f3 key?
;Branch if Equal
;f5 key?
;Branch if Equal
;f7 key?
;Branch if Equal
;exit to default IRQ interrupt routine
;load into X the value found at 145
;only bit 3 set to 0? (\$FB=251=%11111011)
;exit detour if CTRL key is not pressed
;Load X with value of 24 (cursor row)
;Load Y with a zero (cursor column)
;Clear Carry flag for PLOT at \$E50A
;Kernal PLOT jump to 58634, set cursor
; location
;exit detour and resume normal IRQ
; interrupt
;\$CB76-\$CB7B: effect same as BASIC
;IF PEEK(154)<>3 THEN SYS 59953
;i.e., exit if output is not to screen
;prepare for subtraction, SET Carry flag
;subtract 133 from A, so 0=f1, 1=f3,

Fig. 20-23. Continued.


```

; 2=f5, 3=f7
CB7F 0A      ASL      ;Arithmetic Shift Left = binary doubling
CB80 0A      ASL      ;in effect, multiply value in Accumulator
CB81 0A      ASL      ;by 2*2*2*2, or 16 - necessary to calc
CB82 0A      ASL      ;offset address in data table below
CB83 18      CLC      ;prepare for addition, CLear Carry flag
CB84 69 98   ADC  #98   ;add 98 to product now in Accumulator
CB86 85 FB   STA  $FB   ;result to zero page for low-byte process
; at $CB8E
CB88 A9 CB   LDA  #$CB   ;$CB88-$CB8B: location $FC value always
CB8A 85 FC   STA  $FC   ;hi-byte for address calculated at $CB8E
CB8C A0 00   LDY  #$00   ;use Y as counter, beginning at zero
CB8E B1 FB   LDA  ($FB),Y ;Load A with byte at address calculated
CB90 F0 E1   BEQ  $CB73  ;branch to exit if A loads a zero
CB92 20 D2 FF JSR  $FFD2  ;Kernal CHROUT, prints char to screen
CB95 C8      INY      ;increment Y, increase count by one
CB96 D0 F6   BNE  $CB8E  ;branch because byte in A is non-zero
;$CB98-$CBCC are data table bytes
CB98:44 41 54 41 20 00 00 00 ;/DATA@@@
CBA0:00 00 00 00 00 00 00 00 ;/@@@@@@@
CBA8:4C 4F 41 44 22 24 22 2C ;/LOAD"$",
CBB0:38 00 00 00 00 00 00 00 ;/8@@@@@@
CBB8:56 45 52 49 46 59 22 2A ;/VERIFY"*
CBC0:22 2C 38 00 00 00 00 00 ;/",8@@@@
CBC8:4C 49 53 54 00 ;/LIST@

```

Fig. 20-23. Continued.

64 TIMER (page 329)

```

C000 78      SEI
C001 A9 13   LDA  #$13
C003 8D 14 03 STA  $0314
C006 A9 C0   LDA  #$C0
C008 8D 15 03 STA  $0315
C00B 58      CLI
C00C 60      RTS
C00D 29 0F   AND  #$0F
C00F 18      CLC
C010 69 30   ADC  #$30
C012 60      RTS
C013 AD 0B DC LDA  $DC0B
C016 20 0D C0 JSR  $C00D
C019 8D 20 04 STA  $0420
C01C AD 0A DC LDA  $DC0A
C01F 20 0D C0 JSR  $C00D
C022 8D 23 04 STA  $0423
C025 AD 09 DC LDA  $DC09
C028 20 0D C0 JSR  $C00D
C02B 8D 26 04 STA  $0426
C02E AD 0B DC LDA  $DC0B

```

Fig. 20-24.

```

C031 20 6E C0 JSR $C06E
C034 29 01 AND #$01
C036 20 0F C0 JSR $C00F
C039 8D 1F 04 STA $041F
C03C AD 0A DC LDA $DC0A
C03F 20 6E C0 JSR $C06E
C042 29 07 AND #$07
C044 20 0F C0 JSR $C00F
C047 8D 22 04 STA $0422
C04A AD 09 DC LDA $DC09
C04D 20 6E C0 JSR $C06E
C050 29 07 AND #$07
C052 20 0F C0 JSR $C00F
C055 8D 25 04 STA $0425
C058 A9 20 LDA #$20
C05A 8D 1E 04 STA $041E
C05D 8D 27 04 STA $0427
C060 A9 3A LDA #$3A
C062 8D 21 04 STA $0421
C065 8D 24 04 STA $0424
C068 AD 08 DC LDA $DC08
C06B 4C 31 EA JMP $EA31
C06E 4A LSR
C06F 4A LSR
C070 4A LSR
C071 4A LSR
C072 60 RTS

```

Fig. 20-24. Continued.

64 VARIABLE DUMP (page 331)

```

C000 A5 2D LDA $2D
C002 A4 2E LDY $2E
C004 85 FB STA $FB
C006 84 FC STY $FC
C008 A0 01 LDY #$01
C00A B9 FB 00 LDA $00FB,Y
C00D D9 2F 00 CMP $002F,Y
C010 D0 04 BNE $C016
C012 88 DEY
C013 10 F5 BPL $C00A
C015 60 RTS
C016 20 2F C0 JSR $C02F
C019 18 CLC
C01A A5 FB LDA $FB
C01C 69 07 ADC #$07
C01E 85 FB STA $FB
C020 90 02 BCC $C024
C022 E6 FC INC $FC
C024 A9 0D LDA #$0D
C026 20 16 E7 JSR $E716
C029 A5 91 LDA $91

```

Fig. 20-25.

C02B	10	FC		BPL	\$C029
C02D	30	D9		BMI	\$C008
C02F	A0	00		LDY	#\$00
C031	B1	FB		LDA	(\$FB),Y
C033	AA			TAX	
C034	29	7F		AND	#\$7F
C036	20	16	E7	JSR	\$E716
C039	C8			INY	
C03A	B1	FB		LDA	(\$FB),Y
C03C	48			PHA	
C03D	29	7F		AND	#\$7F
C03F	A8			TAY	
C040	20	16	E7	JSR	\$E716
C043	68			PLA	
C044	10	3F		BPL	\$C085
C046	8A			TXA	
C047	30	29		BMI	\$C072
C049	A9	24		LDA	#\$24
C04B	20	AA	C0	JSR	\$C0AA
C04E	A9	22		LDA	#\$22
C050	20	16	E7	JSR	\$E716
C053	B1	FB		LDA	(\$FB),Y
C055	99	FB	00	STA	\$00FB,Y
C058	C8			INY	
C059	C0	05		CPY	#\$05
C05B	D0	F6		BNE	\$C053
C05D	A5	FD		LDA	\$FD
C05F	F0	0C		BEQ	\$C06D
C061	A0	00		LDY	#\$00
C063	B1	FE		LDA	(\$FE),Y
C065	20	16	E7	JSR	\$E716
C068	C8			INY	
C069	C4	FD		CPY	\$FD
C06B	D0	F6		BNE	\$C063
C06D	A9	22		LDA	#\$22
C06F	4C	16	E7	JMP	\$E716
C072	A9	25		LDA	#\$25
C074	20	AA	C0	JSR	\$C0AA
C077	B1	FB		LDA	(\$FB),Y
C079	AA			TAX	
C07A	C8			INY	
C07B	B1	FB		LDA	(\$FB),Y
C07D	A8			TAY	
C07E	8A			TXA	
C07F	20	91	B3	JSR	\$B391
C082	18			CLC	
C083	90	1F		BCC	\$C0A4
C085	A9	20		LDA	#\$20
C087	20	AA	C0	JSR	\$C0AA
C08A	B1	FB		LDA	(\$FB),Y

Fig. 20-25. Continued.

Chapter 20

Fig. 20-25. Continued.

C08C	99	5F	00	STA	\$005F,Y
C08F	C8			INY	
C090	C0	07		CPY	#\$07
C092	D0	F6		BNE	\$C08A
C094	A5	62		LDA	\$62
C096	AA			TAX	
C097	09	80		ORA	#\$80
C099	85	62		STA	\$62
C09B	A0	00		LDY	#\$00
C09D	8A			TXA	
C09E	10	02		BPL	\$C0A2
C0A0	A0	FF		LDY	#\$FF
C0A2	84	66		STY	\$66
C0A4	20	DD	BD	JSR	\$BDDD
C0A7	4C	1E	AB	JMP	\$AB1E
C0AA	20	16	E7	JSR	\$E716
C0AD	98			TYA	
C0AE	D0	05		BNE	\$C0B5
C0B0	A9	20		LDA	#\$20
C0B2	20	16	E7	JSR	\$E716
C0B5	A9	3D		LDA	#\$3D
C0B7	20	16	E7	JSR	\$E716
C0BA	A0	02		LDY	#\$02
C0BC	60			RTS	

128 APPENDER (page 340)

Fig. 20-26.

0B00	A5	2D		LDA	\$2D
0B02	8D	18	0B	STA	\$0B18
0B05	A5	2E		LDA	\$2E
0B07	8D	19	0B	STA	\$0B19
0B0A	38			SEC	
0B0B	AD	10	12	LDA	\$1210
0B0E	E9	02		SBC	#\$02
0B10	85	2D		STA	\$2D
0B12	AD	11	12	LDA	\$1211
0B15	85	2E		STA	\$2E
0B17	60			RTS	
0B18	01	01		ORA	(\$01,X)
0B1A	01	01		ORA	(\$01,X)
0B1C	01	01		ORA	(\$01,X)
0B1E	01	01		ORA	(\$01,X)
0B20	AD	18	0B	LDA	\$0B18
0B23	85	2D		STA	\$2D
0B25	AD	19	0B	LDA	\$0B19
0B28	85	2E		STA	\$2E
0B2A	60			RTS	
0B2B	01	01		ORA	(\$01,X)
0B2D	01	01		ORA	(\$01,X)
0B2F	01	F8		ORA	(\$F8,X)

128 BORDER KILLER (page 233)

0B00	A9	11		LDA	#\$11
0B02	8D	14	03	STA	\$0314
0B05	A9	0B		LDA	#\$0B
0B07	8D	15	03	STA	\$0315
0B0A	A9	FF		LDA	#\$FF
0B0C	8D	50	0B	STA	\$0B50
0B0F	58			CLI	
0B10	60			RTS	
0B11	AD	52	0B	LDA	\$0B52
0B14	8D	51	0B	STA	\$0B51
0B17	AD	19	D0	LDA	\$D019
0B1A	8D	19	D0	STA	\$D019
0B1D	EE	50	0B	INC	\$0B50
0B20	AD	50	0B	LDA	\$0B50
0B23	F0	18		BEQ	\$0B3D
0B25	A9	FF		LDA	#\$FF
0B27	8D	50	0B	STA	\$0B50
0B2A	A9	19		LDA	#\$19
0B2C	8D	11	D0	STA	\$D011
0B2F	A9	F8		LDA	#\$F8
0B31	8D	12	D0	STA	\$D012
0B34	AD	51	0B	LDA	\$0B51
0B37	8D	21	D0	STA	\$D021
0B3A	4C	65	FA	JMP	\$FA65
0B3D	A9	17		LDA	#\$17
0B3F	8D	11	D0	STA	\$D011
0B42	A9	2F		LDA	#\$2F
0B44	8D	12	D0	STA	\$D012
0B47	AD	53	0B	LDA	\$0B53
0B4A	8D	21	D0	STA	\$D021
0B4D	4C	65	FA	JMP	\$FA65
0B50	00			BRK	
0B51	00			BRK	
0B52	00			BRK	
0B53	00			BRK	
0B54	78			SEI	
0B55	20	84	FF	JSR	\$\$\$FF84
0B58	20	81	FF	JSR	\$\$\$FF81
0B5B	20	8A	FF	JSR	\$\$\$FF8A
0B5E	58			CLI	
0B5F	60			RTS	

Fig. 20-27.

128 RESTORE COLORS (page 342)

Fig. 20-28.

```

1400  20 CC FF JSR $FFCC
1403  20 7A 41 JSR $417A
1406  20 8D 41 JSR $418D
1409  20 12 41 JSR $4112
140C  20 38 52 JSR $5238
140F  A9 00 LDA #$00
1411  85 15 STA $15
1413  58 CLI
1414  A9 00 LDA #$00
1416  8D 21 D0 STA $D021
1419  A9 00 LDA #$00
141B  8D 20 D0 STA $D020
141E  A9 00 LDA #$00
1420  85 F1 STA $F1
1422  A2 00 LDX #$00
1424  BD 37 14 LDA $1437,X
1427  20 D2 FF JSR $FFD2
142A  E8 INX
142B  E0 01 CPX #$01
142D  D0 F5 BNE $1424
142F  A9 0D LDA #$0D
1431  20 D2 FF JSR $FFD2
1434  4C 37 4D JMP $4D37
1437  00 BRK

```

128 SIMULTANEOUS TEXT AND GRAPHICS
(page 230)

Fig. 20-29.

```

0B00  78 SEI
0B01  A9 0D LDA #$0D
0B03  8D 14 03 STA $0314
0B06  A9 0B LDA #$0B
0B08  8D 15 03 STA $0315
0B0B  58 CLI
0B0C  60 RTS
0B0D  A9 01 LDA #$01
0B0F  45 D8 EOR $D8
0B11  85 D8 STA $D8
0B13  4C 65 FA JMP $FA65

```

SCREEN SCRAMBLER (page 219)

1300	A9 04	LDA #04
1302	85 FB	STA \$FB
1304	A0 00	LDY #00
1306	84 FA	STY \$FA
1308	84 FC	STY \$FC
130A	B1 FA	LDA (\$FA),Y
130C	C9 20	CMP #20
130E	F0 0F	BEQ \$131F
1310	A2 01	LDX #01
1312	86 FC	STX \$FC
1314	C9 00	CMP #00
1316	D0 02	BNE \$131A
1318	A9 21	LDA #21
131A	AA	TAX
131B	CA	DEX
131C	8A	TXA
131D	91 FA	STA (\$FA),Y
131F	E6 FA	INC \$FA
1321	D0 E7	BNE \$130A
1323	E6 FB	INC \$FB
1325	A5 FB	LDA \$FB
1327	C9 08	CMP #08
1329	D0 DF	BNE \$130A
132B	A5 FC	LDA \$FC
132D	C9 00	CMP #00
132F	D0 CF	BNE \$1300
1331	60	RTS

Fig. 20-30.

Other Languages

*“When you’re lying awake with a dismal headache,
And repose is tabooed by anxiety,
I conceive you may use any language you choose
To indulge in without impropriety.”*

Gilbert and Sullivan
Pirates of Penzance
1879

Because BASIC is built into every Commodore machine, most of us have written BASIC programs. Commercial programs need the speed of machine language to do their job, so most of them are written in machine language with an assembler of one kind or another.

But our Commodore machines will run other languages. This chapter can begin your acquaintance with these all-too-unfamiliar tongues.



A Feast of Languages

While most programming for our Commodores is done in the BASIC and machine language that are built into the computer, many other languages are available.

When programming in these languages, you typically use the Commodore screen editor to prepare your program, giving it line numbers as in BASIC, but using the vocabulary and syntax of whatever language you're writing in. Your program is then processed by one or more other programs, either memory resident or on disk, to produce executable and SAVEable code in the second language.

The program you type in is called the *source code*. The one prepared later is called the *object code*.

Here's a list of the most popular alternate languages:

Assembly Language This is a brother to machine language, because the code they produce is exactly the same. With assembly language, however, the programmer has access to many programming tools that machine language lacks by itself. All but the very simplest "machine language" programming is really done in "assembly language" with an assembler program; some people use the two terms interchangeably.

BASIC (Compiled) Commodore's built-in BASIC is a so-called "interpreted" language. This means that program statements are converted to machine language one-by-one each time the program is run. The various "compiled" BASICs have the familiar vocabulary and syntax, but after the program is written it is subjected to further processing; it is *compiled*. This step converts the BASIC into a machine language program which is the program you actually run. When this compiled program is executed, the computer doesn't spend time converting each statement from BASIC. Compiled BASIC is faster than the interpreted variety, but you cannot LIST the compiled program.

C (That's not a misprint, it's the name of a language.) The C language is much in vogue among professional programmers. It's a powerful language with the major advantage that its programs are transportable from one make of computer to another. C is somewhat difficult to learn, but many people think it's worth it. It's very big among Amiga programmers.

COBOL The computer that printed your paycheck is probably programmed in COBOL, the business world's most popular language. COBOL is very effective at processing large numbers of repetitive business-type transactions; that's why it's so popular on mainframes.

COMAL This interesting language has Commodore devotees all around the world. A more complete description appears in "COMAL Explained," below.

FORTRAN Here's a cousin of BASIC, with a roughly similar syntax and vocabulary. FORTRAN's great power is its ability to handle complicated mathematical calculations. It's a favorite language of scientists.

FORTH Available in several versions, FORTH is a great language for those who like the unusual. It's highly customizable, and you program in it by defining your own new commands.

LOGO An easy-to-learn language that has been taught in elementary schools. Its so-called "turtle graphics," in which lines are drawn by a "turtle" crawling around the screen, are popular with young programmers.

Modula-2 An improvement on Pascal. Easier. Lets you write your programs in modules, which are then linked together into a whole.

Pascal Many colleges require their students to program in Pascal. It's a highly structured language, requiring the programmer to plan his program before he writes it. The rigid structuring makes it easy for other people to look at a program and understand it. If you're a teacher, you like that feature. If you're a student, then maybe you don't.

*Mike Rivers
West Chester, Pennsylvania*



COMAL Explained

COMAL is an interesting programming language for the 64, 128, Amiga and PET, as well as other computers such as the Macintosh, IBM PC, IBM PS/2 and CP/M machines. It was developed in Denmark as a first language for home programmers, but has evolved to the point where it's used by professionals in many countries.

The language is similar to Pascal, C and Modula-2. Programs written on one machine can be run on any other, so long as machine-specific enhancements haven't been used. Commodore versions are available for as little as \$9.95!

The U.S. branch of the COMAL Users Group publishes an 80-page newsletter (23 issues to date), and has dozens of COMAL-specific books and disks at reasonable prices.

For more information, send a self-addressed business-size envelope with two stamps to:

COMAL Users Group, U.S.A., Ltd.
5501 Groveland Terrace
Madison, WI 53716

△☆ **Compiled COMAL**

The latest 64 disk-loaded version of COMAL, called Power Driver, has a compiler option. If you use the compiler, you can pass around copies of your COMAL programs without requiring users to have COMAL, or even to know that you used COMAL to write the programs.

*Len Lindsay
Madison, Wisconsin*

△☆ **COMAL Turtle Graphics Tip**

When doing turtle graphics, the drawing is much quicker if you turn off the turtle image with the `hideturtle` command.

*Captain Comal
Copenhagen, Denmark*

△☆ **COMAL File Copier**

This program written in COMAL 2.01, will copy PRG, SEQ or USR files of any length from drive 8 to drive 9. When entering the filename, you must end it with a comma and the one-letter file type designator (P, S or U).

```
10 INPUT "FILENAME: ": name$
20 OPEN FILE 2,"0:"+name$,READ
30 OPEN FILE 3,"2:"+name$,WRITE
40 WHILE NOT EOF(2) PRINT FILE 3:GET$(2,20000),
50 CLOSE
```

*Ray Carter
Las Cruces, New Mexico*

△☆ **LOGO and Print Shop**

You can use LOGO to create pictures for use with Print Shop. After creating your LOGO picture on the screen, type `SAVEPICT"filename"`, where *"filename"* is the name you give to your picture.

This saves the picture as a 33-block hi-res file named `"filename.PIC1"` and a 5-block color file named `"filename.PIC2"`. The first one will load into Screen Magic when you select the Get Screen option. After loading it, you can alter it, add text, and load and save the altered picture at will.

*Barrett Stridiron
Charlotte Amalie, Virgin Islands*



Super C Memory Access

If you use the C64 V2 version of Super C to access memory which can be bank switched (video chip, SID, etc.), you should do this with a machine language routine. The Super C operating system has control of the memory map, and while it is in control of the machine, it will use memory in its own way. Your machine language routine should first save the contents of location \$0001 and then set this location to get the required map. Perform your required operations, and before exiting your routine, restore the original value to location \$0001.

*Jim Oldfield, Jr.
Abacus Software*



Super C Bug

There's a minor bug in the STRCMP() and STRNCMP() functions in some versions of Super C 128 V3. These functions may only be called around 120 times, and then the computer will hang. This has been fixed in V3 version #3.02. Avoid heavy use of these functions if you have the older version, or purchase an update from Abacus.

*Rob Lun
East Grand Rapids, Michigan*



Super Pascal File and Variable Names

Abacus Software's Super Pascal prefers that you keep file names and variable names limited to eight characters. If you use longer variable names, you may get strange results.

*Gene Traas
Wyoming, Michigan*



Super Pascal File Safety

When you look at the directory of the Super Pascal disk, you'll notice that the names of source files have an 'S' and a back-arrow as a prefix. This is not a requirement of the language, but it's a good convention to follow.

In Pascal, you're required to supply a program name in the source code; the compiler uses it as the filename under which to save the object code. If you have unwisely used that name when saving your source code, the compiler will overwrite it.

*Jim D'Haem
Kentwood, Michigan*

The CP/M Operating System

*“Our little systems have their day;
They have their day and cease to be;
They are but broken lights of thee,
And thou, O Lord, art more than they.”*

Alfred, Lord Tennyson
In Memoriam
1850

The advent of “IBM compatibility” and MS-DOS have brought CP/M system development to the twilight of its days. Nevertheless, there are thousands of high-quality CP/M programs already in existence, proven through widespread use and just waiting to be run on your 128.

Here are some tips on using them.



What Is CP/M?

The 128 has a sometimes mysterious mode called CP/M, about which we read very little. In CP/M mode, your computer behaves very much unlike a Commodore, and very much like an IBM or clone. CP/M is an operating system, or a group of programs that coordinates all the workings of your computer.

In its normal mode, your 128 uses a built-in operating system designed specifically for the 128. The 128 operating system lets you do things like formatting disks, displaying directories, loading programs, and changing colors. Because it's built into the computer, you seldom even notice it.

CP/M is a totally different operating system that is loaded in from disk. In the early 1980's it was used by many manufacturers; Commodore, Apple and Radio Shack were notable exceptions. The common use of CP/M allowed interchange of software between computers, although many hardware incompatibilities remained.

When the IBM PC was introduced, it created a hardware standard for other computer makers to follow. Most of them switched from CP/M to MS-DOS, a twin of the IBM's PC-DOS operating system. Commodore and Apple remained as exceptions, but Radio Shack switched to MS-DOS.

With the CP/M disk you got with your 128, your computer can run under the CP/M operating system. When it does so, it requires totally different commands for formatting disks, displaying directories, loading programs and other similar operations.

In CP/M mode, you can run much of the CP/M applications software that was so popular before IBM standardization. Much of it is very good, and most of it is inexpensive or free.

Even if the applications software doesn't interest you, you can gain a lot by becoming familiar with CP/M. It has the look and feel of MS-DOS, so if you learn how to work with CP/M, much of your skill will transfer to the IBM world.

It's a good idea to order the CP/M book and disk that are offered in the 128 System Guide. For under \$20, you get a very complete manual and two useful utility disks. Without the manual, it's next to impossible to puzzle out CP/M.

There are many commercial books on CP/M, as well. When looking at them, remember that your computer runs CP/M + version 3.0, which is quite different from earlier versions.

☆ **Locating CP/M Software**

There's a huge amount of CP/M software available, but it's often not easy to find. Here are two good sources: Poseidon Electronics, 103 Waverley Place, New York, NY 10011, (212) 777-9515. FOG, P.O. Box 3474, Daly City, CA 94015. Poseidon is a company, FOG is a user group.

*Stewart Terrier
Alton, Illinois*

☆ **Kaypro CP/M Programs on the 128**

There are many public domain programs for the Kaypro CP/M computers that can be used with the 128 and 1571 drive. When you load a Kaypro disk, the prompt KAYPRO IV will appear on the screen. If you press the Cursor Right key, the disk will load correctly.

*Barrett Stridiron
Charlotte Amalie, Virgin Islands*

☆ **128 CP/M Reboot**

You don't have to use the Reset button or power switch to reboot in the 128's powerful CP/M mode. Just press the Control and Enter keys at the same time. (Be sure to use the Enter key on the numeric keypad, and not the Return key on the main keyboard.)

If your CP/M system disk is not in Drive A, the computer will reboot into BASIC.

*Jay D. Locklear
Fayetteville, North Carolina*

☆ **Changing CP/M Screen Width**

When loading the CP/M + systems disk, I occasionally forget to put the 40/80 column switch in the proper position. A simple solution avoids having to reboot the system. To go into 80-column mode, just type this at the A prompt:

```
DEVICE CONOUT: = 80 COL
```

If you wish to go the other way, use 40COL instead.

*Herbert Nelson
Chicago, Illinois*

 ☆ **CP/M Disk Swapping**

You can easily use files on one disk to operate on files on another, even if you only have one drive. The key is to use drive E, the so-called virtual drive.

For example, if the DIR file is on the disk in drive A, you can use it to examine another disk by entering:

```
DIR E:[FULL]
```

When you're prompted to insert the other disk (disk E), just insert it into your drive and press Return.

*Tom Malcom
Mount Pleasant, Iowa*

 ☆ **128 CP/M and Printers**

When using CP/M programs with a 1525 or plug-compatible printer, several problems arise. The worst is the destruction of the double quotation mark character; another is that many CP/M programs attempt underlining by overprinting the ASCII underline character, which comes out as text overstruck with back-arrows.

The problem can be fixed if you have a 1571 or 1581 drive and the two extra disks you order from DRI. (The order form is in the 128 System Guide).

As always, make backup copies of all the disks you will use, and store the originals in a safe place outside your computer room. Also keep a bootable disk with your present CP/M system, in case something goes wrong.

When all your backups have been made, copy the files from both sides of the SOURCE disk to a single double-sided disk. Using your word processor or a text editor such as CP/M's ED.COM, modify the file CXPRINTE.ASM in the following way:

Find the routine CONVT:, which should be on or around line 76. It should read:

```
?convt:
    ani    7fh
    cpi    cr
    jrz    set$msb
    cpi    '"'
    jrz    is$quote
    cpi    '@'
```


Change it so it reads:

```
?convt:
    ani    7fh
    cpi    cr
    jrz    set$msb
    cpi    ' '
    jrz    is$und
    cpi    '@'
```

Note that the underline character (used to replace the double quote in the cpi statement) is on the back-arrow key at the upper left of the keyboard.

Now find the routine IS\$QUOTE:, which should be at or around line 110. It should read:

```
is$quote:
    mvi    a,27h ;convert to tick
    ret
```

Change this to read:

```
is$und:
    mvi    a,0a4h ;convert to CBM underline
    ret
```

Save these changes to disk, then close the file. If your editor makes a backup file, erase it.

Now follow the directions in the file MAKESYS.DOC on the SOURCE disk, substituting your new CXPRINTE.ASM file. Find something else to do for a while, because the process, even with two disk drives, takes over half an hour to complete.

Finally, copy the CPM + .SYS file to a new disk, and also copy over the CCP.COM file from your current boot disk. Boot from the new disk, and try printing a file.

If all went well, you now not only have the capability to print the double quote, but also you can underline, even with the 1525! Most CP/M word processors, including WordStar and Perfect Writer, will allow you to underline in this way. Be sure you keep a copy of your original system, though. Then if a bug shows up years later, you can still reboot from the backup system.

*James T. Peebles
Texarkana, Arkansas*

Software Potpourri

*“The time has come,” the Walrus said,
“To talk of many things:
Of shoes—and ships—and sealing wax—
Of cabbages—and kings—”*

Lewis Carroll
Through the Looking-Glass
1872

This chapter contains the software tips that don't fit comfortably elsewhere. As you can see, that takes in a lot of ground. Also, as you can see, that ground is very fertile.



Bytes

A byte is the elementary unit of storage in Commodore computers and many others. For practical purposes, a byte can be thought of as equivalent to one printed character. The word “Commodore,” then, with nine letters, would take nine bytes of storage space on a disk or in main memory.

A kilobyte, or K, is 2^{10} bytes, or 1024 bytes. It gets its name from the Greek prefix meaning one thousand, though strictly speaking it’s a little larger than that.

A megabyte, often used in measuring the storage capacity of hard disks, is nominally a thousand K. More accurately, it’s 1024K, or 1,048,576 bytes.

There are about 2000 characters on a double-spaced typewritten sheet of paper, so all the text on such a page could be stored in about 2K of memory or disk space. One disk in a 1541 drive holds about 175K bytes, or the equivalent of about 90 double-spaced typewritten pages. Because the 1571 drive is double-sided, its disks can hold 350K bytes of information, or about the same as that in 175 typewritten pages.

The smaller hard disk drives commonly used in IBM PC’s and clones will hold 20 megabytes of information, or five times the number of characters as in the King James version of the Bible.



Naming Variables

It’s useful to develop your own conventions for naming variables in your programs. Since I developed that habit, it’s much easier for me to debug long programs, and easier to go back into old programs and figure what’s going on.

I always reserve A\$ as the variable for GET statements and other places where I can disregard the variable after checking its value. Because I never use A\$ for anything that will be useful later in the program, I know it’s always safe to use it in a new section.

Also, I never use J for anything but the variable in a FOR...NEXT loop, and I always use J as the first variable in such a loop. If other variables are needed inside the loop, or in a nested loop, my practice is to use K, L and M. If the nesting gets too complicated, I use J1, J2, J3 and so on. By the way, I never use the letter ‘I’ as a variable name, because it’s too easily confused with the numeral ‘1’ (one).

When selecting other variable names, I try to choose them to be somewhat meaningful or mnemonic. I limit them to two letters, because they’re easier to type than longer names, and because I haven’t found

much additional mnemonic value in the longer names. To me, DN\$ is just as easy as DISKNAME\$, and I don't have to worry about the confusion if I later set up a DI\$ or DISK\$.

*Lucy S. Terrier
Alton, Illinois*



Temporary Line Deletion

If you want to see the effect of removing a certain line from your program, use the Inst key to open up three spaces after the line number, then type REM into the opened space. When you press Return, the line will be changed to a remark. If the change doesn't give the effect you want, you can reinstate the line by using the Del key to delete the REM, then pressing Return again. If the change improves your program, delete the line by typing its number then pressing Return.

*Stewart Terrier
Alton, Illinois*



LIST Stopper

When working on a program, I often want its screen listing to stop at a particular point. Many times, for example, programs have large numbers of DATA statements at the end, and I'm working on the part of the program just before the DATA statements. I have no need to list the DATA statements, but I'm keenly interested in the code that appears before them.

I borrow a trick from the program protector's handbook, and use a REM [SHFT L] to stop the listing where I want it. When the computer encounters such a line, the listing terminates immediately, even though the program runs normally in every respect.

*John Ettinger
Warren, Ohio*



Automatic Line Listing

When writing or improving a program, I find myself repeatedly listing certain groups of lines. This becomes much easier when I don't have to type the LIST command in full. How do I do it? In one of the three simple ways described below.

The easiest way is available only to owners of the 128, Plus/4 or any computer whose function keys print information on the screen. Just enter something like this from the keyboard in direct mode:

KEY 1, "LIST 2000-2150" + CHR\$(13)

From then on, you can LIST lines 2000-2150 just by pressing the F1 key. The + CHR\$(13) at the end of the line simulates pressing the Return key after the LIST command is printed to the screen. If you prefer to press Return yourself, just leave that part out of your key definition. Of course, you should substitute your own line numbers for the ones in this example.

When you no longer have much need to list your original range of lines (2000-2150 in our example), you can redefine KEY 1 to print a different range. You can put different line number ranges on other function keys by executing similar statements with different numbers after the KEY command.

You can also use this system to get printer listings of all or part of your program. Just enter something like this:

```
KEY 2, "OPEN 4,4 : CMD 4 :"
```

Now pressing the F2 key will put those commands on the screen, with the cursor blinking right after the final colon. To list your program to the printer, type LIST and hit the Return key. If you've put a LIST command on another function key, just press that key at this point, and your listing will come out on the printer!

If you don't have easily programmable function keys, you can use one of these two alternate methods for getting easy listings. The first is to start your program with one or more lines like this:

```
0 GOTO 100
1 LIST 2000-2150
2 LIST 3000-3999
3 OPEN 4,4 : CMD 4 : LIST 2000-2150
4 etc.
100 REM REGULAR PROGRAM STARTS HERE
```

When you run the program in the normal way, line 0 will bypass your special additions. To get a listing of lines 2000-2150, just RUN 1. To get a printed listing, RUN 3. You can extend this as far as you'd like, and it can save a lot of repeated typing.

On the 64, executing a LIST from program mode will terminate program execution, so you don't need an END statement at the end of your special "listing lines." But other computers, notably the 128, do not terminate execution after a LIST. If yours is one of them, you'll need to add an END or STOP at the end of each of the listing lines.

The third way to get automatic listing is a variation on the one above. Instead of the lines shown there, use these:

```
0 GET A$ : IF A$="" THEN 0
1 IF A$="A" THEN LIST 2000-2150
2 IF A$="B" THEN LIST 3000-3999
3 IF A$="C" THEN OPEN 4,4 : CMD 4 : LIST 2000-2150
4 etc.
100 REM REGULAR PROGRAM STARTS HERE
```

Now when you run the program, nothing happens until you press another key. When you press A, B, C or any of the keys specified in your IF statements, the appropriate lines will be listed. If you press any other key, the program will fall through to line 100, and the regular part of the program will be executed.

I've used all three methods at various times, and they've saved me thousands of seconds and keystrokes. I hope you can use them as productively.

*Harold H. Hogg
Ross Township, Pennsylvania*



Speed Testing

When trying to speed up a program, it's common to read the TI variable at the start and end of the section you're working on. A simple subtraction tells how many "jiffies" the section took to execute. I've found it useful to GOSUB to a "beep" routine just before reading TI at the start of the test, and immediately after reading it at the end.

Though the TI subtraction gives an accurate measurement of elapsed time, the pause between the two beeps often gives a more understandable measurement.

*Tracey S. Sander
Pittsburgh, Pennsylvania*



Faster Execution

There are many ways to make your BASIC programs run faster. Here are some of the more popular ones:

- Delete all unnecessary spaces and REMs from your program.
- Put frequently executed subroutines at the beginning of the program, using a GOTO to skip over them when the program is run.
 - Put DATA statements at the beginning of the program, right after the subroutines.
 - If certain variables are referenced frequently, use a line early in the program to assign them their initial values, even if these are

zero. The earlier a variable is defined, the faster BASIC can reference it.

- When assigning a zero value to a variable, or when using zero as an argument in a function, type a decimal point instead of the numeral zero. The computer treats the decimal point as a zero, but reads it considerably faster.
- In FOR...NEXT loops, type the NEXT without the variable name.
- In a series of IF statements, make the first one the one with the greatest probability of being true.
- Replace IF statements with ON...GOTOs where possible.

*Timothy Sullivan
Brooklyn Park, Minnesota*



Sensing Special Keys

When designing a program that uses keyboard input, it's useful to be able to sense special keys like Shift, Control and Commodore. It's really quite easy to do, because there's a memory location that has various bits set depending on which of these keys are pressed.

The location is 653 for the 64 and VIC, 211 for the 128, and 1347 for the Plus/4. When the Shift key is pressed, it contains a 1. When the Commodore key is pressed, it contains a 2. And when the CTRL (Control) key is pressed, it contains a 4.

It holds those values only as long as the indicated keys are actually in the down position.

You can also detect combinations by adding the values. For example, [CTRL] + [SHFT] = 5 and [SHFT] + [CMDR] = 3.

*Jason Barney
Renton, Washington*



Detecting Disabled Stop Key

If you've disabled the Stop key, you can still detect it and have your program take the appropriate action. The secret is to use a GET statement to check for a CHR\$(3), which is the character generated by the disabled Stop key.

*Skynyrd
Quantum Link*



Machine Type Detector

There are many ways for a program to detect which Commodore computer it is running on, but I like PEEK(65534) the best. The results will be as shown below.

64: PEEK(65534) = 72
128: PEEK(65534) = 23
Plus/4: PEEK(65534) = 255

These were checked on my own machines, and are believed to be accurate for all. But there's a chance that ROM variations could cause them to change.



Editing SEQ Files

Programmers often store data in sequential files that are later read by their programs. Unfortunately, if a SEQ file becomes corrupted, finding and correcting the problem can be a major headache.

Compounding the difficulty is that many errors are created by bugs in the program you are writing. So you need to analyze the SEQ file so you can analyze the mistakes in your program!

One of the easiest ways to look at your SEQ files and to make corrections is to use your word processor. Most will allow you to load a sequential file, letting you examine the items one at a time until you discover the error.

*Charles W. Lewis, Jr.
Fitzgerald, Georgia*



Efficient Numeric Files

When numbers are saved to disk files, the operating system first converts them to strings, which are normally separated by CHR\$(13)'s so they can be retrieved by the INPUT# statement.

But if your numbers are integers between 0 and 255, there's a much more efficient way to save them. If the number to be saved is N, for example, use:

```
PRINT#2,CHR$(N);
```

to save it, and use:

```
GET#2,A$ : N = ASC(A$)
```


to retrieve it. The savings in disk space are phenomenal, because each number is saved as a single byte on the disk, without any extra delimiters.

Art Kohn
Duke University
Durham, North Carolina

○△☆ Correcting Input

Figure 23-1 illustrates a useful technique for correcting input errors without unnecessary retyping. After inputting your data, the routine asks you if everything is correct. If it isn't, your cursor jumps to the first line of your original input. If necessary, retype the entry or simply press Return to move the cursor to the next entry.

Joseph R. Charnetski
Plains, Pennsylvania

```
100 REM CORRECTING INPUT - J.CHARNETSKI
110 PRINT "[CLEAR]"
120 INPUT "FIRST NAME";N1$
130 INPUT " LAST NAME";N2$
140 INPUT " TELEPHONE";TN$
150 INPUT "[DOWN]ALL CORRECT (Y OR N)";A$
160 IF LEFT$(A$,1) <> "Y" THEN PRINT "[HOME]":GOTO 120
170 PRINT:PRINT N1$:PRINT N2$:PRINT TN$
```

Fig. 23-1.

○△☆ Character Format Conversions

In the non-Commodore computer world, the ASCII code is often used for transferring text from place to place. For example, word processors often save text to disk in ASCII, and BASIC uses ASCII codes in CHR\$ statements. In the Commodore world, we use a somewhat different code, often called PETASCII (from its original use in Commodore's PET computer), for these purposes. When POKEing characters to the screen, Commodore computers use yet another code, often called the Screen Poke Code.

All three codes work in the same general way: each displayable character has a unique numeric code that can be easily handled internally by the computer. But, unfortunately, there are significant differences in the codes. For example, a lowercase "a" is represented by 97 in ASCII, 65 in PETASCII, and 1 in Screen Poke. Because programmers frequently

Chapter 23

```
100 REM CHARACTER FORMAT CONVERSIONS
110 REM LOUIS F. SANDER
120 :
130 REM FNPS(X) PETASCII TO SCREEN POKE
140 REM FNSP(X) SCREEN POKE TO PETASCII
150 REM FNPA(X) PETASCII/ASCII AND BACK
160 :
170 DEF FN A1(X)= X-161-33*(X<255)-64*(X<192)-32*(X<160)+32*(X<96)
    -64*(X<64)
180 DEF FN A2(X)= ABS(X*(X<128 OR X>159))-32*(X>127 AND
    X<160)-(32-X)*(X<32)
190 DEF FN A3(X)= X-64*(X<32 OR X>95)-32*(X>63 AND X<96)
200 DEF FN A4(X)= X AND 127
210 DEF FN A5(X)= X+(X*(X>127))-126*(X>127)
220 DEF FN A6(X)= X-32*(X>64 AND X<91)+32*(X>96 AND
    X<123)+128*(X>192 AND X<219)
230 DEF FN PS(X)= FN A1(FN A2(X))
240 DEF FN SP(X)= FN A3(FN A4(X))
250 DEF FN PA(X)= FN A5(FN A6(X))
```

Fig. 23-2.

need to convert data from one of these formats to another, there's a real need for computer algorithms to make the conversion.

Figure 23-2 contains User defined Functions that can be used to convert any of these formats to any other. Once your program executes lines 170-250, you can use the functions to make the conversions.

Be careful when typing lines 170, 180 and especially 220; if you insert extra spaces, these lines may exceed 80 characters in length. Lines longer than 80 characters cannot be directly entered into a 64.

The functions have been named to make it easy to know their purpose. FNPS converts codes from PETASCII to Screen Poke. FNSP converts from Screen Poke to PETASCII. FNPA converts from PETASCII to ASCII, and as an added bonus also converts ASCII to PETASCII. Say, for example, that you want to convert PETASCII 97 to its Screen Poke equivalent and to store the result in variable ZJ. Just execute ZJ = FNPS(97).

When using these functions, you should keep some important points in mind:

- When converting text, the Commodore should be in upper and lowercase mode, not graphics mode.
- PETASCII codes from 0-31 and 128-159 have no corresponding screen pokes. FNPS converts them to 32's (spaces).

- Screen Poke codes greater than 127 (for RVS characters) have no corresponding PETASCII codes. FNPA converts them to the PETASCII codes of their unreversed equivalents.
- PETASCII and ASCII differ greatly in the meanings of control codes 0-31, and somewhat in characters 92, 94, 95, 96 and 123-127. FNPA makes no conversions on any of those codes.
- ASCII has no codes greater than 127, while PETASCII goes all the way up to 255. Where PETASCII codes above 127 can be converted to ASCII equivalents, FNPA makes the conversion. It changes unconvertible PETASCII characters to ASCII 126's (tildes).

*Louis F. Sander
Pittsburgh, Pennsylvania*



64 Program Protection

Here's a technique that hides a 64 program so it cannot be listed. It also disables the Stop and Stop/Restore keys. The technique is permanent, so if you ever want to list or edit your program, save a copy before protecting it. Also, the protected program cannot use lines 0, 1 or 2. The first step is to add this special line 2 to the program:

```
2 POKE 43,PEEK(2049) : POKE 44,PEEK(2050) : POKE 808,234
```

Then replace all END's and exit points with:

```
POKE 43,1 : POKE 44,8 : POKE 808,237
```

Next, add this line. It's a very unusual one because the Inserts, Deletes, etc. are performed after two quotation marks have been typed:

```
1 REM " "[LEFT,INST20,DEL10,SPACE10,RIGHT]
NO FAIR PEEKING! [SHFT L]
```

Finally, enter this in direct mode:

```
POKE 2051,255 : POKE 2052,255
```

*John Ettinger
Warren, Ohio*



Defeating Program Protection

Many simple program protection schemes involve something that interferes with listing the program to the screen. A REM followed by a shifted L will do this, as will various REMs with DELete characters.

Very often these schemes will be revealed if you list your program to the printer, rather than to the screen.

*William F. Sander
Marietta, Ohio*



Deleting 64 Program Lines

Here's a shortcut to deleting blocks of program lines on the 64. Execute a POKE 774,0 to modify your computer's LIST routine. The modified routine lists line numbers only, without the accompanying program material. After the POKE, just LIST the lines you want to delete, then press Return over each of the numbers.

POKE 774,26 to restore your computer to normal, then LIST your program to verify that the deleted lines are gone.

The Plus/4 equivalents of these commands are POKE 774,187 and POKE 774,110.

*Robert B. Cook
Braintree, Massachusetts*



64 Line Number Suppression

If you want to avoid the line numbers in a listing, just POKE 22,35. With POKE 22,25 the line numbers will be shown again. The same POKEs, believe it or not, will work on the Plus/4!

*Christian A. Wentzensen
Asuncion, Paraguay*



64 POP Routine

Here is a simple routine to clean out any unresolved GOSUB/RETURNS or FOR/NEXT loops. I do not recommend the practice of writing programs that need this. But if you must, you must! Just type this in a line in your program where you want to POP:

```
POKE 38,104:POKE 39,104:POKE 40,76  
:POKE 41,126:POKE 42,166:SYS 38
```

After this line executes, all FOR/NEXT loops and GOSUBs will act as if they had finished or been RETURNed from.

*Kermit R. Woodall
Glen Allen, Virginia*

**128 Reset Changer**

If you have a 128 and are going to do a lot of work in 64 mode, do this in 128 mode:

BANK 1 : POKE 65528,77 : POKE 65529,255 : SYS 65357

This changes the operating system vector that determines which mode will be active after a reset. After it is executed, pressing the Reset button will put your computer into 64 mode, without your having to hold down the Commodore key. You must turn the computer off and back on to restore the vector and go into 128 or CP/M mode.

*Daihung Ho
East Moline, Illinois*

**128 Line Counter**

Have you ever wondered how many lines it took to make that program? If you own a 128 or Plus/4, all you have to do is enter **RENUMBER 1,1** and list the program. The last line number is your answer.

If you save the program before renumbering it, the saved version will retain the original line numbers.

*VictorP6
Quantum Link*

Related Tips in Other Chapters

Here are some additional tips like those in the potpourri:

Chapter 7	Easier Printer Control
Chapter 19	Structured REMs
Chapter 30	Lottery Assistant
	Dice Machine

Programming the Video Display

*“Lawyers and painters
can soon change white to black.”*

Danish Proverb

So can programmers, and in the world of Commodore computing, we're fortunate to have sixteen beautiful colors, 316 different keyboard characters, two types of high-resolution graphics, and the ability to display them on anything from a TV set to the highest-tech video monitor.

Whatever video display you have, you can use these tips to make it shine.

○△☆ Centering Text

Here's how to tell how much to TAB a word or phrase to center it on the screen. Count the letters in your text and subtract it from the number of characters on a screen line, then divide the total by two.

This will do it in BASIC, assuming that your text is M\$:

```
PRINT TAB((40 - LEN(M$))/2)
```

Because of BASIC's way of doing math, you must put parentheses around the `40 - LEN(M$)`.

*Steve Dodd
Edmonton, Alberta
Canada*

△☆ Shaky Computing

Has your computer ever had a nervous breakdown? It might seem that way with this:

```
10 FOR J = 0 TO 15 : POKE 53270,J : NEXT : GOTO 10
```

By eliminating the GOTO, you can use it as a special effect in BASIC games or other programs using the text screen.

For an even stranger effect, change the 15 in the FOR . . . NEXT loop to a 31.

*James Frisa, III
Cincinnati, Ohio*

△☆ Screen Saver

The program in Fig. 24-1 provides a fast and easy way of saving and displaying text screens. It saves a 64 or 128 text screen as a four-block disk file which you can reload at any time.

When the program is run, you'll be prompted to input a filename for the screen to be saved. Enter the filename and press Return, and the screen will clear with a flashing cursor in the home position. At this point, just type the screen you want to save, using any of the keyboard keys except Return and the color keys. When you are finished, press Return and the screen will be saved under the name you previously assigned it.

Once the screen has been saved, you can display it by typing:

```
LOAD "filename",8,1
```

Chapter 24

```
100 PRINT"[CLEAR,RVS,SPACE5]SCREEN SAVER - MICHAEL MCCURDY[SPACE5]"
110 PRINT"[DOWN]THIS SAVES A TYPED SCREEN AS A DISK FILE"
120 PRINT"[DOWN2]WHEN YOU ENTER THE FILENAME, THE SCREEN
130 PRINT"WILL CLEAR.[SPACE2]TYPE WHAT YOU WANT TO SAVE,"
140 PRINT"BUT DO NOT PRESS RETURN UNTIL YOU ARE"
150 PRINT"FINISHED."
160 PRINT"[DOWN]WHEN FINISHED, PRESS RETURN AND YOUR"
170 PRINT"SCREEN WILL BE SAVED."
180 PRINT"[DOWN]FILENAME:";:OPEN 1,0:INPUT#1,S$
    :IF LEN(S$)>16 THEN 100
190 PRINT"[CLEAR]";:INPUT#1,A$:CLOSE 1
200 OPEN 2,8,1,S$:PRINT#2,CHR$(0);CHR$(4);
210 FOR J=1024 TO 2023:PRINT#2,CHR$(PEEK(J));:NEXT
220 CLOSE 2
```

Fig. 24-1.

where “filename” is the name under which you have saved the screen.

*Michael McCurdy
Ann Arbor, Michigan*



“Color” on a Monochrome Monitor

Commodore’s 16 colors are just a keypress away, but they are useless to those of us without color monitors—or so we think. All we see are different shades of gray, green or amber, depending on the phosphor in our monitor.

It’s useful to know which colors show up as various shades of gray. From dark to light, here’s how they appear on my own monochrome: black, blue, brown, dark gray, red, orange, purple, light blue, medium gray, light red, green, cyan, light gray, yellow, light green, white. Sometimes adjacent shades will show up almost identically, and different monitors may cause minor variations in the order of the shades presented, but for most purposes this order is perfectly correct.

This knowledge can be useful in many ways, such as in causing two sprites to collide, flash brightly then fade out.

*Rick Root
Saline, Michigan*



Fancy Printing

Figure 24-2 shows three subroutines that can add excitement to your screen displays. The first makes letters drop from the sky before lining up in the center of a screen line. The second zooms them in from the

left, and the third makes them burst out from the middle of the screen. All you have to do is set the desired text equal to M\$, then call the appropriate subroutine. The listing includes a demonstration of all three routines.

You can use one, two or three of the routines in your own programs. To make them work with 80-column displays, just change the 40's to 80's in the three definitions of variable TA. In "dropping letters from the sky," variable TD determines which screen line they will settle on. The dropping letters will erase the lines above this one on the screen, except for line #1.

Thong Nguyen
Ft. Walton Beach, Florida

```

10 PRINT"[CLEAR,RVS,SPACE5]FANCY PRINTING - THONG NGUYEN[SPACE6]"
20 M$="COMMODORE COMPUTERS"
30 GOSUB 100:PRINT
40 GOSUB 200:PRINT
50 GOSUB 300:PRINT:PRINT
60 END
70 :
100 REM ** LETTERS DROP FROM THE SKY
110 TD=10:REM SCREEN LINE TO PRINT ON
120 L=LEN(M$):TA=(40-L)/2-1:FOR J=1 TO L:M1$=MID$(M$,J,1)
    :PRINT"[HOME]"
130 IF M1$=" " THEN 150
140 FOR K=1 TO TD-2:PRINT TAB(TA+J)M1$"[UP]":PRINT TAB(TA+J) " ":NEXT
150 PRINT TAB(TA+J)M1$:NEXT
160 RETURN
170 :
200 REM ** LETTERS ZOOM IN FROM LEFT
210 L=LEN(M$):TA=(40-L)/2-1:FOR J=L TO 1 STEP-1:M1$=" "+MID$(M$,J,1)
220 IF M1$<>"[SPACE2]" THEN FOR K=1 TO J:PRINT TAB(TA+K-1)M1$"[UP]"
    :NEXT
230 NEXT:PRINT
240 RETURN
250 :
300 REM ** LETTERS EXPLODE FROM CENTER
310 IF LEN(M$)/2<>INT(LEN(M$)/2) THEN M$=M$+" "
320 L=LEN(M$):TA=(40-L)/2:TB=L/2:TC=TA+TB:FOR J=1 TO TB:TC=TC-1
330 PRINT TAB(TC);LEFT$(M$,J);TAB(TB);RIGHT$(M$,J);"[UP]"
340 FOR K=1 TO 20:NEXT:NEXT
350 RETURN

```

Fig. 24-2.



Changing Screen Color

If you're writing a program that allows the user to change the screen color, the routine in Fig. 24-3 can be very useful. When executed, it advances the screen color by one number. The good part is that the screen and character colors are never allowed to be the same!

```

100 REM CHANGING SCREEN COLOR
101 REM BY KERMIT R. WOODALL
102 REM ADVANCES THE SCREEN COLOR, BUT
103 REM WILL NOT MAKE IT THE SAME AS
104 REM THE CHARACTER COLOR.
105 :
110 X=646:REM 646 FOR 64, 241 FOR 128
120 POKE 53281,(PEEK(53281)+1 AND 15)+ABS((PEEK(53281)+1 AND
    15)=(PEEK(X) AND 15))

```

Fig. 24-3.

Basically, a simple trick of computer logic is used to check the screen color and keep it from being the same as the character color. Don't worry about how it works—it just does!

Be careful when typing line 110; if you insert extra spaces, its length may exceed the 80-character limit for typing into a 64.

*Kermit R. Woodall
Glen Allen, Virginia*



Changing Character Color

Figure 24-4's routine operates like the one in Fig. 24-3, except that it works on the character color. When you execute it, the character color increments, but is never allowed to be the same as the screen color.

```

100 REM CHANGING CHARACTER COLOR
101 REM BY KERMIT R. WOODALL
102 REM ADVANCES THE CHARACTER COLOR
103 REM BUT WILL NOT MAKE IT THE SAME
104 REM AS THE SCREEN COLOR.
105 :
110 X=646:REM 646 FOR 64, 241 FOR 128
120 POKE X,(PEEK(X)+1 AND 15)+ABS((PEEK(53281) AND 15)=(PEEK(X)+1
    AND 15))

```

Fig. 24-4.

As with the previous program, be careful when typing line 110; if you insert extra spaces, its length may exceed the 80-character limit for typing into a 64.

Kermit R. Woodall
Glen Allen, Virginia



Computer Shake

The little program in Fig. 24-5 "shakes" your computer screen for a second or two, then lets it go. It's another example of what can be done with a very few bytes of programming code.

Arbi Sookazian
Herndon, Virginia

```
10 FOR J=0 TO 255:POKE 53270,J:NEXT:POKE 53270,200
   :REM COMPUTER SHAKE - A.SOOKAZIAN
```

Fig. 24-5.



Screen Scrambler

Figure 24-6 is a machine language routine that decrements the screen code values of any displayed characters until the 40-column screen is cleared. It works anywhere in 64 or 128 memory that there are 50 free bytes. To relocate it, just change the variable AD to the new address.

```
10 REM SCREEN SCRAMBLER - BATCHELOR
15 REM RELOCATABLE ML FOR 64 & 128
20 AD=4864:FOR J=0 TO 49:READ K:POKE AD+J,K:NEXT
25 DATA 169,004,133,251,160,000,132,250
30 DATA 132,252,177,250,201,032,240,015
35 DATA 162,001,134,252,201,000,208,002
40 DATA 169,033,170,202,138,145,250,230
45 DATA 250,208,231,230,251,165,251,201
50 DATA 008,208,223,165,252,201,000,208
55 DATA 207,096
60 :
100 REM DEMONSTRATIONS OF THE EFFECT
110 FOR J=1 TO 1000:PRINT"[L. RED,RVS] ";:NEXT:SYS AD
120 FOR J=1 TO 60:PRINT"[L. BLUE,RVS]012345678987654321";:NEXT
   :SYS AD
130 FOR J=1 TO 24:PRINT"[YELLOW,RVS]GHIJKLMNOPQRSTUVWXYZYXWVUTSRQPO
   NMLKJIHG":NEXT
140 SYS AD
```

Fig. 24-6.

Lines 100-140 are not part of the routine itself, but are included to demonstrate some interesting applications.

Chris Batchelor
Big Spring, Texas



Screen Framer

Here's another old-timer, this one from an early newsletter called *The PET Paper* (they liked cute names in those days). The program in Fig. 24-7 puts a decorative frame around your 40-column screen, composed of the character specified by F\$ in line 5010.

```
5000 REM SCREEN FRAMER - THE PET PAPER
5010 F$=" [CMDR +] "
5020 PRINT "[CLEAR] ";
5030 FOR J=1 TO 39:PRINT F$;:NEXT
5040 PRINT "[HOME] "
5050 FOR J=1 TO 23
5060 PRINT F$;TAB(38);F$
5070 NEXT
5080 FOR J=1 TO 39:PRINT F$;:NEXT
5090 PRINT "[HOME] "
```

Fig. 24-7.

You'll notice that the frame stops one character short of the right-hand screen border. The easiest fix for this is to make the screen and border the same color; even though the frame doesn't change, it looks like it's centered on the screen.

Other fixes are easier said than made, but they can, of course, be done. The main difficulty is that the cursor jumps to the next line as soon as a character is printed in the rightmost column, and it makes the computer think it's printing on a wrapped-around 80-column screen line. (You'll see the effect of this when you try printing inside the frame.)

Good solutions can be based on doing a Cursor Left, then printing an "insert" character to shove the F\$ over to the edge. With that and the rest of your bag of tricks, you should be able to get an almost-perfect frame.

The "almost" involves the rightmost position on the bottom screen line. When you PRINT a character there, the screen scrolls upward and you lose the top of your border. I usually solve the problem by POKEing the appropriate character into that position.

**64 Screen Bars**

Figure 24-8 gives you an interesting way to highlight your 64 screen display. After you run it, executing a **SYS 49152** will put colorful bars onto the screen for several seconds. As you can see from the REMs, you can change the duration of the display and the vertical size of the bars by poking special values. As presented here, the screen is cleared before the bars are displayed. You can eliminate the screen clearing by executing a **POKE 49153,0**. (It returns when you **POKE 49153,147**).

```

10 REM 64 SCREEN BARS - MATTHEW LEWIS
20 REM POKE 49153,0 FOR NO CLEAR SCREEN
30 REM POKE 49153,147 TO RESTORE CLR
40 REM POKE 49199,TIME (2 IS NORMAL)
50 REM POKE 49215,BAR HEIGHT (2 NORMAL)
60 REM SYS 49152 TO ACTIVATE THE BARS
70 FOR J=0 TO 82:READ K:POKE 49152+J,K:NEXT
71 DATA 169,147,032,074,192,162,000,160
72 DATA 000,169,000,141,032,208,141,033
73 DATA 208,238,010,192,173,010,192,201
74 DATA 017,208,005,169,000,141,010,192
75 DATA 032,051,192,232,224,255,208,225
76 DATA 238,096,192,173,096,192,201,002
77 DATA 208,215,096,200,192,255,208,251
78 DATA 238,097,192,173,097,192,201,002
79 DATA 208,241,169,000,141,097,192,096
80 DATA 234,234,032,210,255,169,000,141
81 DATA 096,192,096

```

Fig. 24-8.

In all cases, your POKEs must be executed before the **SYS 49152**.

*Matthew Lewis
Christ Church, Barbados*

**64 Color Flipper**

The program in Fig. 24-9 lets you instantaneously change the colors of all text on the screen, just by executing a **SYS 53049**. It works differently with different 64 ROM versions, so you may have to experiment with the numbers you **POKE** into 53086 and 53087 to get the colors you want. On some machines, the colors you end up with depend on the original color of the characters when printed to the screen.

*Richard Miu
Address Unknown*

Chapter 24

```
10 PRINT"[CLEAR,RVS,SPACE5]64 COLOR FLIPPER - RICHARD MIU[SPACE5]"
20 PRINT"[DOWN]SYS 53049 TO FLIPS CHARACTER COLORS."
30 PRINT"[DOWN]POKE COLORS INTO 53086 AND 53087."
40 POKE 53086,0:POKE 53087,1:REM WHT/BLK
50 FOR Y=1 TO 37:READ Z:CS=CS+Z:NEXT:RESTORE
60 IF CS<>6031 THEN PRINT"[DOWN]DATA ERROR":STOP
70 FOR J=53049 TO 53085:READ K:POKE J,K:NEXT
80 DATA 173,094,207,077,095,207,141,096
81 DATA 207,169,000,133,251,169,216,133
82 DATA 252,162,004,160,000,177,251,077
83 DATA 096,207,145,251,200,208,246,230
84 DATA 252,202,208,239,096
```

Fig. 24-9.



64 F-Key Colors

There's no more need to enter dozens of tedious POKEs to come up with suitable color combinations—the program in Fig. 24-10 makes it easy as F1, F2, F3!

```
10 PRINT"[CLEAR,RVS,SPACE4]64 F-KEY COLORS - SHUKRI BERISHA[SPACE4]"
20 FOR J=828 TO 922:READ K:POKE J,K:CS=CS+K:NEXT
30 IF CS<>10460 THEN PRINT"ERROR IN DATA STATEMENTS!":STOP
40 SYS 828:PRINT"[DOWN]PRESS F1, F3 & F5 TO CHANGE COLORS."
50 PRINT"[DOWN]STOP/RESTORE DISABLES THE FEATURE,"
60 PRINT"SYS 828 RE-ENABLES IT.[DOWN2]"
61 DATA 120,169,073,141,020,003,169,003
62 DATA 141,021,003,088,096,165,197,201
63 DATA 004,240,050,201,005,240,035,201
64 DATA 006,208,050,205,153,003,240,045
65 DATA 238,134,002,173,134,002,160,000
66 DATA 153,000,216,153,000,217,153,000
67 DATA 218,153,000,219,200,240,022,076
68 DATA 100,003,205,152,003,240,014,238
69 DATA 033,208,076,137,003,205,151,003
70 DATA 240,003,238,032,208,165,197,141
71 DATA 151,003,141,152,003,141,153,003
72 DATA 076,049,234,000,000,000,000,000
```

Fig. 24-10.

Once the program has been run, those three keys change the border, background and character colors, respectively. This feature is particularly useful when you're testing color combinations for use in your own programs. I use it as a routine in larger programs, allowing the user to select his or her own colors.

Shukri Berisha
Missisauga, Ontario
Canada

 △ **64 Color SYS**

Here's another easy way to set the border, background and character colors on your 64. While the previous program is great for choosing colors from the keyboard, the one in Fig. 24-11 is best for setting predetermined colors from within a program.

```

10 PRINT"[CLEAR,RVS,SPACE6]64 COLOR SYS - DAILAH HODGE[SPACE7]"
20 FOR J=828 TO 854:READ K:POKE J,K:CS=CS+K:NEXT
30 IF CS<>3012 THEN PRINT"ERROR IN DATA STATEMENTS!":STOP
40 PRINT"[DOWN]SYS 828,BO,BK,CH TO SET THE 64'S BORDER,"
50 PRINT"BACKGROUND AND CHARACTER COLORS.[DOWN2]"
51 DATA 032,155,183,138,072,032,155,183
52 DATA 138,072,032,155,183,138,141,134
53 DATA 002,104,141,033,208,104,141,032
54 DATA 208,096,000
  
```

Fig. 24-11.

Once Fig. 24-11's routine has been executed, SYS 828,BO,BK,CH will set your border, background and character colors, respectively. BO, BK and CH are the numbers from 0-15 that specify colors on the 64.

Dailah Hodge
Address Unknown

 △ **64 Easy Windows**

Windows, windows, windows! Everywhere you look, computers seem to be sporting windows. Even the 64 has turned into a windowing machine, with many great programs written to let it show windows on the screen.

What are those windows, anyway? In most cases, a window is nothing more than a rectangular block of text that pops up on an existing screen, usually containing some information that the user wants to see for a while, then remove. The window is usually smaller than the screen it appears on, so the effect is that of a Post-It note temporarily pasted over the material already on the screen.

The program in Fig. 24-12 lets you make windows of your own, without committing the gobs of memory that commercial windowing programs require.

Once lines 100-158 have been executed, SYS 828 will save your existing screen to the empty RAM at locations 49152 upward. You can then print on the screen, erase it, draw some sort of window, or whatever you'd like to do for your temporary 'window' effect. There's no need to make your window smaller than the whole screen.

```

100 PRINT"[CLEAR,RVS,SPACE4]64 EASY WINDOWS - JEREMY HUBBLE[SPACE5]"
110 PRINT"[DOWN]SYS 828 TO SAVE SCREEN"
120 PRINT"[DOWN]SYS 858 TO RETRIEVE SCREEN"
130 PRINT"[DOWN2]WORKING..."
140 FOR J=828 TO 887:READ K:POKE J,K:CS=CS+K: NEXT
150 IF CS<>7006 THEN PRINT"ERROR IN DATA STATEMENTS!!":STOP
151 DATA 160,000,185,000,004,153,000,192
152 DATA 185,000,005,153,000,193,185,000
153 DATA 006,153,000,194,185,232,006,153
154 DATA 232,194,200,208,229,096,160,000
155 DATA 185,000,192,153,000,004,185,000
156 DATA 193,153,000,005,185,000,194,153
157 DATA 000,006,185,232,194,153,232,006
158 DATA 200,208,229,096
159 :
1000 REM DEMONSTRATE WINDOWING
1010 PRINT"[UP]TO DEMONSTRATE WINDOWING,"
1020 PRINT"PRESS ANY KEY..."
1030 GET A$:IF A$=""THEN 1030
1040 SYS 828: REM SAVE THE SCREEN
1050 PRINT"[CLEAR,DOWN2]THIS IS PRINTED IN THE 'WINDOW'."
1060 PRINT"[DOWN]LIKE THE FIRST SCREEN, IT COULD"
1070 PRINT"CONTAIN ANYTHING AT ALL."
1080 PRINT"[DOWN]PRESS ANY KEY TO CLOSE THE 'WINDOW'"
1090 PRINT"AND RECALL THE ORIGINAL SCREEN..."
1100 GET A$:IF A$=""THEN 1100
1110 SYS 858:REM RECALL THE SCREEN

```

Fig. 24-12.

When you're finished with the window, SYS 858 recalls the former screen from 49152, writing over anything you've printed to the screen in the meantime.

Lines 1000-1110 are a demonstration of the Easy Windows principle. You can remove them and lines 100-130 when you incorporate 64 Easy Windows into programs of your own. (As a kindness to the creator, you might preserve the author's name in a REM.)

Jeremy Hubble
Belton, Texas



64 PRINT@

Some non-Commodore versions of BASIC have a statement named PRINT@, which lets the programmer easily place the cursor at any position on the screen.


```

10 REM 64 PRINT@ - SUMIT AGGARWAL
20 REM TO USE: SYS 850,ROW,COLUMN FOLLOWED BY THE PRINT STATEMENT
30 FOR J=850 TO 865:READ K:POKE J,K:NEXT
40 DATA 032,155,183,138,072,032,155,183
50 DATA 104,170,164,101,024,076,240,255

```

Fig. 24-13.

Figure 24-13 gives you the power to do the same on your 64. Once you have executed line 30, you can place the cursor by executing:

SYS 850,row,column

where *row* and *column* are the numbers of the appropriate screen location (the top row is numbered zero, as is the leftmost column). In the program, *row* and *column* can be numbers, variables or expressions.

Sumit Aggarwal
New Delhi, India



64 Screen Effect

To see an interesting special effect, run the little program in Fig. 24-14. Press Stop to stop it. If you add other lines to the end of the program, they'll execute when the effect is stopped.

David Brown
Address Unknown

```

10 REM 64 SCREEN EFFECT - DAVID BROWN
20 SA=49152 :REM RELOCATABLE START ADDR
30 FOR J=0 TO 24:READ K:POKE SA+J,K:CS=CS+K:NEXT
40 IF CS<>3953 THEN PRINT"ERROR!":STOP
50 DATA 200,140,033,208,140,032,208,162
60 DATA 094,202,016,253,165,203,201,063
70 DATA 208,238,165,203,201,064,208,250
80 DATA 096
90 SYS SA

```

Fig. 24-14.



64 Border Fireworks

Figure 24-15's program puts a distinctive sparkling pattern into the 64's normally uninteresting screen border. As soon as the user presses a key, the border returns to its previous color. I use the routine, along with an on-screen message, to let the user know that the computer is waiting for some action from him.

```

10 REM 64 BORDER FIREWORKS - TOM MORROW
20 REM SYS679 STARTS, KEYPRESS STOPS
30 FOR J=679 TO 693:READ K:POKE J,K:NEXT
40 DATA 172,032,208,202,142,032,208,165
50 DATA 198,240,248,140,032,208,096

```

Fig. 24-15.

To prepare the fireworks, include the lines from Fig. 24-15 somewhere at the start of your program. After they've been executed, a SYS 679 will set the fireworks off from any part of the program, as many times as you'd like.

Tom Morrow
Oak Park, Illinois



128 Auto-Centering

This interesting text-centering technique for the 128 uses a built-in, but mostly forgotten, feature of BASIC 7.0: the PRINT USING statement. It's illustrated by the program in Fig. 24-16. Line 20 determines the character width of the current window and builds string CN\$ to equal that width. The leading equals sign tells BASIC to center the information on the line.

```

10 REM 128 AUTO-CENTERING - JOSEPH GILL
20 CN$=" ":FOR J=1 TO RWINDOW(1):CN$=CN$+"#":NEXT
100 :
110 PRINT"[CLEAR,DOWN]AUTO-CENTER DEMONSTRATIONS:[DOWN2]"
120 PRINT USING CN$;"THIS IS EASILY CENTERED!!"
130 PRINT
140 PRINT USING CN$;"SO IS THIS:"
150 PRINT
160 PRINT USING CN$;"A","MAN","THREW","BASEBALLS","WHILE","I",
    "WATCHED","HIM"

```

Fig. 24-16.

Lines 100-160 merely demonstrate the effect in our example. Only line 20 is needed to implement it.

When you want to center text in the window, just use

```
PRINT USING CN$;"text"
```

where "text" is whatever you want to center, expressed literally or as a string variable.

More than one text line can be centered with the same command by separating the text lines with commas, as shown in line 160.

If you change window width, repeat line 20. If you do this often, you may want to put line 20 in a subroutine.

*Joseph P. Gill
Address Unknown*



128 Screen POKEs

Don't listen to those who say you can't POKE characters to the 128's 80-column screen. The 128 has a screen editor routine to send bytes to the 80-column chip. To use it in BASIC, simply type the following:

```
SYS 52684,byte,31
```

where *byte* is the screen poke code for the character you want to appear on the screen. It will appear at the current cursor position, and the cursor position will advance by one when you send the byte to the chip.

You can set the cursor position anywhere you want by using this code:

```
SYS 52684,byte1,18 : SYS 52684,byte2,19
```

where *byte1* and *byte2* make up the screen address where the character is to appear. The home position has *byte1* = 0 and *byte2* = 0; incrementing *byte2* increments the cursor position by one, while incrementing *byte1* increments the cursor position by 256. For the lower right-hand corner of the screen, *byte1* = 7 and *byte2* = 207.

The same SYS call can be used to determine a character's color or its attributes such as underline, reverse field, blinking or graphics upper and lowercase. To do this, determine the position of the character you want to affect, then add 8 to *byte1* in the expression above. Execute the resulting expression, then follow it with `SYS 52684,byte,31` where *byte* is a number that will determine the attributes. The attribute table is too large to print here, but it's easy to determine it experimentally.

You can PEEK the screen by using this (don't forget either one of the two commas):

```
SYS 52698,,31 : RREG X
```

After execution, the variable *X* will hold the value of the character at the cursor position. If the cursor isn't where you want it, you can set its position by the techniques described above.

*Michael Kimsal
Mt. Clemens, Michigan*

```
10 REM 128 80-COLUMN LINE SPACE CHANGER
15 REM ALLEN S. CHEUNG
20 BANK 15:POKE 54784,8:POKE 54785,3:POKE 54784,9:POKE 54785,15
   :POKE 54784,23:POKE 54785,15
```

Fig. 24-17.

☆ **128 80-column Line Space Changer**

The line of code in Fig. 24-17, executable in direct or program mode, makes an interesting change to the 128's 80-column display. When it is executed, the characters shrink to half their normal height, with the missing height showing up as space between the lines.

The modified display looks great, so I always execute this line before booting up my word processor.

Pressing the Stop/Restore combination will return the characters to their normal size.

*Allen S. Cheung
Clairton, Pennsylvania*

☆ **128 Simultaneous Text and Graphics**

One of the problems in programming 128 graphics is that you can't see your typing and its results at the same time. If you use one of the split screen graphics modes you can get around this to some extent, but it's not very satisfactory for graphics that cover the entire screen, or for graphics that take more than a few lines of programming.

The program in Fig. 24-18 lets you have your cake and eat it too! It switches the screen rapidly between text and graphics modes, so that both displays are visible at once. There's some flickering on the screen, but it's a small price to pay for the convenience of seeing everything together.

You can disable the program by hitting Stop and Restore, and you can activate it again by executing a SYS 2816.

The CIRCLE statement in line 210 is only for demonstration purposes. You can eliminate it from the version you save for use as a utility.

*Marco A. Gonzalez Hagelsieb
Guadalajara, Mexico*

```
100 REM 128 SIMULTANEOUS TEXT & GRAPHICS
110 REM MARCO HAGELSIEB
120 :
130 REM STOP/RESTORE KILLS
140 REM SYS 2816 REACTIVATES
150 :
160 FOR J=2816 TO 2837:READ K:POKE J,K:CS=CS+K:NEXT
   :IF CS=2226 THEN 180
170 PRINT"DATA ERROR. CHECK TYPING.":END
180 GRAPHIC 1,1:GRAPHIC 0,1
190 PRINT"NOW YOU CAN HAVE TEXT AND GRAPHICS"
   :PRINT"ON SCREEN AT THE SAME TIME!!"
200 SYS 2816
210 CIRCLE 1,160,100,85,70
220 DATA 120,169,013,141,020,003,169,011
230 DATA 141,021,003,088,096,169,001,069
240 DATA 216,133,216,076,101,250
```

Fig. 24-18.



80-column Flashing and Underlining

On the 128's 80-column screen, underlining is initiated by printing a CTRL B or CHR\$(2), and canceled by a CHR\$(130). Similarly, flashing text is started by CTRL O or CHR\$(15), and stopped by CHR\$(143). The CTRL B and CTRL O can be entered within a PRINT statement just by pressing the appropriate key combinations when the cursor is inside quotes. But there aren't any key combinations that will give the CHR\$(130) or CHR\$(143), so turning off these features requires lots of manipulation of the PRINT statements.

If you are using a lot of flashing and underlining, it may pay you to redefine some function keys, like this:

```
KEY 4, CHR$(130)
KEY 6, CHR$(143)
```

Now, if your cursor is inside quotes, you can cancel the feature by pressing the appropriate function key.

If you really want to get fancy, put CHR\$(2) on the F3 key and CHR\$(15) on the F5 key. Then you can use the shifted and unshifted function keys to turn the features on and off.

*Eugene L. Pizzetta
Revere, Massachusetts*

☆ **Pausing 128 Graphics**

It's sometimes desirable to pause the creation of a hi-res graphics screen in mid-program, especially when developing software. Unfortunately, the manual doesn't tell you how to do it, or even if it can be done.

My trick for this is to turn on the trace option by entering TRON before running the program. With this option enabled, you can pause the creation of the hi-res screen simply by pressing the No Scroll key.

*J. C. Vollmer
Prior Lake, Minnesota*

☆ **128 Symmetrical Scaling**

One of the problems with the 128's wonderful graphic commands is that there are different scales on the X and Y axes. (If you draw a circle, you'll see what I mean—it's more like an egg). The SCALE statement can be used to correct this distortion.

For my 13" diagonal monitor, SCALE 1,2327,2000 makes both axes symmetrical. After executing that statement, CIRCLE 1,1000,1500,300,300 is a perfect circle, and BOX 1,900,900,1200,1200,,1 is a perfect square. If your own monitor shows something different, adjust the 2327 up or down until you get what you want. My friend's 1902 worked well with SCALE 1,2500,2000.

*Glen Ritchie
Bismarck, North Dakota*

☆ **128 Screen Mode Swapper**

Many 128 programs work best in either 40 or 80 column mode, with the other mode inferior or unacceptable. A well written program will first check to see if the computer is in the proper screen mode, and if it isn't, will switch to the correct mode.

The short routine in Fig. 24-19 checks memory location 215 to determine which screen mode is active, then uses the screen swapper kernal routine at 65375 to make the switch if necessary.

*John R. Howell
Houston, Texas*

```

100 REM 128 SCREEN MODE SWAPPER - HOWELL
110 REM PEEK(215)=0 IS 40 COL MODE
120 REM PEEK(215)=128 IS 80 COL MODE
130 IF PEEK(215)=128 THEN PRINT "[CLEAR]
      SWITCH MONITOR TO 40 COL MODE" : SYS 65375

```

Fig. 24-19.

☆ 128 Sprite Rotator

Figure 24-20 can speed up your sprite editing sessions. It lets you design sprite 1, then rotates that sprite in 90 degree steps and puts the result into sprites 2, 3 and 4.

When you run it, you are immediately presented with a SPRDEF screen. (Your 128 System Guide contains detailed instructions for using this automated sprite editing screen. Look in the index under SPRDEF.) Select sprite 1 and design your sprite, being careful not to use the three rightmost columns (there's no place to rotate them to). When you're finished designing your sprite, press Shifted Return, and the 128 will prompt you for another sprite number. Press Return at this point, and wait for 128 Sprite Rotator to do its work.

```

100 N$="128 SPRITE ROTATOR-JOHN MAHOWALD"
110 FOR J=1 TO 8:SPRITE J,0:NEXT
120 PRINT"[CLEAR]";N$:PRINT"[DOWN2]PRESS SPACEBAR TO BEGIN..."
130 GET A$:IF A$<>CHR$(32) THEN 130
140 DIM A(21,21):COLOR 0,1:COLOR 4,1:SPRITE 1,0,2:PRINT"[CLEAR]"
      :SPRDEF:SPRSV 1,S$
150 GRAPHIC 1,1:GSHAPE S$,1,1,0
160 CHAR 0,1,4,N$:CHAR 0,1,6,"PLEASE STAND BY (30 SECONDS)..."
170 FOR Y=1 TO 21:FOR X=1 TO 21:LOCATE X,Y:A(X,Y)=RDOT(2):NEXT:NEXT
180 FOR Y=1 TO 21:FOR X=1 TO 21:DRAW A(Y,22-X),X+30,Y:NEXT:NEXT
190 FOR Y=1 TO 21:FOR X=1 TO 21:DRAW A(X,22-Y),X+60,Y:NEXT:NEXT
200 FOR Y=1 TO 21:FOR X=1 TO 21:DRAW A(Y,X),X+90,Y:NEXT:NEXT
210 FOR N=0 TO 3:SSHAPE S$,N*30+1,1,N*30+22,22:SPRSV S$,N+1:NEXT
220 SPRDEF:MOVSPR 5,160,100:SPRSV 1,5:SPRITE 5,1,2
230 PRINT N$:INPUT"[DOWN]WANT TO USE JOYSTICK (Y/N)";A$
      :IF A$<>"Y" THEN 270
240 PRINT"[DOWN]JOYSTICK MOVES SPRITE, FIRE BUTTON QUILTS"
250 J=JOY(2):IF J>0 AND J<9 THEN SPRSV (J+1)/2,5:MOVSPR 5,J*45-45#5
      :ELSE MOVSPR 5,0#0
260 IF J>127 THEN GRAPHIC 0:ELSE GOTO 250
270 INPUT "[DOWN]SAVE SPRITES (Y/N)";A$:IF A$<>"Y" THEN END
280 F$="":INPUT "[DOWN4]FILENAME";F$:IF F$>"" THEN BSAVE ""+F$,B0,
      P3584 TO P3840

```

Fig. 24-20.

When it finishes rotating and transferring the sprites, the program will put you back into SPRDEF mode, where you can examine all four sprites and work on them some more if you'd like. Exit SPRDEF just as you did above, and the program will give you a chance to move your sprites around using a joystick plugged into Port 2.

Finally, the program lets you save the sprites under any name you choose. To use them later in your programs, just execute:

```
BLOAD "filename",B0,P3584
```

where "filename" is the name under which you saved them.

*John Mahowald
Minnetonka, Minnesota*



128 Screen Explosions

There are many different and interesting ways to simulate an explosion on a computer screen. The subroutine in Fig. 24-21 makes use of low resolution graphics, drawing multicolored lines emanating from the center of the screen.

```
100 REM 128 SCREEN EXPLOSIONS - TIESS
110 REM THIS LINE REPEATS THE EFFECT
120 L=80:S=8:DO:GOSUB 10020:LOOP
130 :
10000 REM 128 SCREEN EXPLOSIONS - TIESS
10010 REM SET L AND S BEFORE CALLING
10020 COLOR 4,1:COLOR 0,1:COLOR 1,3
10030 COLOR 2,10:COLOR 3,8
10040 GRAPHIC 3,1:LOCATE 80,100
10050 FOR A=0 TO 360 STEP S
10060 WIDTH INT(RND(0)*2)+1
10070 C=INT(RND(0)*3)+1
10080 DRAW C,RDOT(0),RDOT(1) TO 80,100
10090 LOCATE +INT(RND(0)*L);A
10100 NEXT
10110 RETURN
```

Fig. 24-21.

The routine is executed by setting variables L (length of line) and S (step angle between lines), then executing a GOSUB 10020. In our listing, line 120 does that for you automatically, then repeats it in an endless loop.

Notice that the explosion has a different appearance each time the subroutine is called. Changing the values of L and S will change the appearance even more.

Robert J. Tiess
Middletown, New York

☆ **128 Border Killer**

The program in Fig. 24-22 is pretty amazing. Consuming only 96 bytes, it turns off your top and bottom borders. With the borders off, you can effortlessly display sprites in the once forbidden zone. Not only that, but because it's interrupt driven, it can remain in the background while other programs are running.

If you want to use the graphics screen then you must first enable it before typing SYS 2816 to activate this program. To disable the program and the graphics screen, enter SYS 2900.

```

10 REM 128 BORDER KILLER-J. GUTTERIDGE
20 REM SYS2816 ENABLES, SYS2900 KILLS
30 REM POKE2898,BKGD:POKE2899,BORDER
40 FOR A=2816 TO A+95:READ B$
50 CK=CK+DEC(B$):POKE A,DEC(B$):NEXT
60 IF CK<>10526 THEN PRINT"DATA ERROR":END
70 DATA A9,11,8D,14,03,A9,0B,8D
71 DATA 15,03,A9,FF,8D,50,0B,58
72 DATA 60,AD,52,0B,8D,51,0B,AD
73 DATA 19,D0,8D,19,D0,EE,50,0B
74 DATA AD,50,0B,F0,18,A9,FF,8D
75 DATA 50,0B,A9,19,8D,11,D0,A9
76 DATA F8,8D,12,D0,AD,51,0B,8D
77 DATA 21,D0,4C,65,FA,A9,17,8D
78 DATA 11,D0,A9,2F,8D,12,D0,AD
79 DATA 53,0B,8D,21,D0,4C,65,FA
80 DATA 00,00,00,00,78,20,84,FF
81 DATA 20,81,FF,20,8A,FF,58,60

```

Fig. 24-22.

Making the pokes in line 30 will change the color of your screen and of the "invisible" border area.

Joel Gutteridge
Drexel Hill, Pennsylvania

○☆ **128 End Writer**

Figure 24-23 adds a visually interesting ending to any 128 program by creating over-size characters on a 40-column hi-res screen. It prints the words "The End" on an upward slope in the center of the screen. Just like Hollywood.

Chapter 24

```
60000 REM 128 END WRITER - J.CHARNETSKI
60010 X=104:Y=112:GRAPHIC 2,1,16:CHAR,0,24,"THE END"
60020 FOR J=1 TO 56:SSHAPE E$,J+2,191,J,199:GSHAPE E$,X+J*2,Y-J:NEXT
60030 END
```

Fig. 24-23.

The program works on the Plus/4, but only if you remove the comma and the 16 from the GRAPHIC statement.

*Joseph R. Charnetski
Plains, Pennsylvania*

Related Tips in Other Chapters

Chapter 6, "Video Displays", has many tips on video display hardware.

These tips also relate to the display:

Chapter 30	Sprite Graph Maker
Chapter 32	Sprite Finder
	128 Hi-res Helper

Programming Music and Sound

“Music hath charms to soothe a savage breast...”

William Congreve
The Mourning Bride
1697

Commodore computers contain some sophisticated musical instruments, as most of us have heard. But playing those instruments is more difficult than we would hope, especially on the 64.

This chapter offers preprogrammed sound effects, little pieces of music, and tools with which to make more of the same. I hope you find them soothing.



Buzzer Sound

Figure 25-1 is a very simple buzzer, such as might be used to signal a wrong answer. It's hard to get 64 sound with any less programming than this.

*Lucy S. Terrier
Alton, Illinois*

```
100 REM BUZZER SOUND - LUCY S. TERRIER
110 FOR J=1 TO 10:POKE 54296,15:FOR K=1 TO 5:NEXT:POKE 54296,0
120 FOR K=1 TO 5:NEXT:NEXT
```

Fig. 25-1.



Better Buzzer

Figure 25-2 shows what can be added to the simple routine above. (Notice that line 110 is identical in both programs.) It adds some tones to the buzzer, lessening its harshness significantly. For more variations, in line 105 change the values after the commas in POKE S,10 and POKE S + 1,40. The new values can be anything between 0 and 255, but they should not both be zero at once. You can also change the POKE S + 4,33 in that line to POKE S + 4, 17 or POKE S + 4, 129.

*Chris Trautman
Delta, British Columbia
Canada*

```
100 REM BETTER BUZZER - CHRIS TRAUTMAN
105 S=54272:POKE S+6,240:POKE S,10:POKE S+1,40:POKE S+4,33
110 FOR J=1 TO 10:POKE 54296,15:FOR K=1 TO 5:NEXT:POKE 54296,0
115 FOR K=1 TO 5:NEXT:NEXT
120 POKE S+4,128
```

Fig. 25-2.



Bell Sound

The routine in Fig. 25-3 gives a clear tinkling bell. It's another simple one, with a very realistic sound.

*Anna Mae Hertzler
Boynton Beach, Florida*

```
200 REM BELL SOUND - ANNA MAE HERTZLER
210 S=54272:POKE S+1,112:POKE S+5,10:POKE S+6,9:POKE S+24,15
220 POKE S+4,17:POKE S+4,16
```

Fig. 25-3.



Bell Tone

To get a very nice bell sound on your 64, just execute the program in Fig. 25-4. Like many bells, this one has an interesting history.

When writing my own software on the 64, I've frequently needed a simple bell sound to signify an error or to tell the user that the computer has finished a job. But being an inexperienced sound programmer, my attempts to create a pleasing bell were rather unsuccessful.

The 128, on the other hand, has a built-in bell tone that is easily called from the keyboard or by a program. I used the 128's built-in Machine Language Monitor to examine the ROM routine that produces this nice sound, then translated it into this:

```
10 REM BELL TONE - RAY ROBERTS
20 REM DUPLICATES 128'S CHR$(7) BELL
30 POKE 54296,21:POKE 54277,9:POKE 54278,0
40 POKE 54273,48:POKE 54276,32:POKE 54276,33
```

Fig. 25-4.

You can use the program as is, or make it into a subroutine. It works identically on the 64 and the 128.

Ray Roberts
Gainesville, Florida



Helicopter Sound

The routine in Fig. 25-5 will make you think Airwolf is hovering outside your window. To make the sound last a longer or shorter time, change the value of D in line 310.

Greg Long
Hillsboro, Oregon

```
300 REM HELICOPTER SOUND - GREG LONG
310 D=100:REM DURATION OF SOUND
320 S=54272:FOR J=0 TO 23:POKE S+J,0:NEXT:POKE S+24,15:POKE S+6,240
330 POKE S+4,17:FOR J=1 TO D:FOR K=9 TO 0 STEP-1:POKE S+1,K:NEXT K
:NEXT J
```

Fig. 25-5.



Beep Routine

Figure 25-6 will give one or more beeps, depending on how many times line 440 is called as a subroutine. (Line 420 calls it twice in our

example.) Once line 410 has been executed, just GOSUB 440 to get a beep.

Ronald C. Newbery, Sr.
Cold Spring, New York

```
400 REM BEEP ROUTINE - RON NEWBERY
410 S=54272:POKE S+24,15:POKE S,20:POKE S+1,20:POKE S+2,10
    :POKE S+3,10
420 POKE S+6,128:FOR J=1 TO 2:GOSUB 440:NEXT:END
430 REM SUBROUTINE TO CALL FOR A BEEP
440 POKE S+4,65:FOR K=1 TO 90:NEXT:POKE S+4,0:RETURN
```

Fig. 25-6.



Explosion Maker

Here's a very flexible sound routine. By changing the values of D and F in lines 510 and 520 of Fig. 25-7, you can get anything from a rifle shot to an avalanche. Have fun exploring the world of explosions!

Mitchell Ross
Delta, Ohio

```
500 REM EXPLOSION MAKER - MITCHELL ROSS
510 D= 9:REM DURATION (0-15)
520 F= 7:REM FREQUENCY (0-255)
530 S=54272:FOR L=1 TO 24:POKE S+L,0:NEXT
540 POKE S+1,F:POKE S+5,16+D:POKE S+24,15
550 POKE S,240:POKE S+4,129
```

Fig. 25-7.



European Siren

The routine in Fig. 25-8 makes a sound like those sirens you hear in spy movies. It's great for using as a subroutine, or for showing someone the amazing sound capabilities of Commodore machines.

```
100 REM EUROPEAN SIREN - MICHAEL GATTO
110 N=5 :REM NUMBER OF SIREN BLASTS
120 S=54272:V=S+24:AD=S+5:SR=S+6:W=S+4
130 POKE V,15:POKE AD,0:POKE SR,240:POKE W,33
140 FOR J=1 TO N
150 POKE S,52:POKE S+1,43:FOR D=1 TO 300:NEXT
160 POKE S,75:POKE S+1,34:FOR D=1 TO 300:NEXT
170 NEXT J
180 POKE W,32:POKE V,0
```

Fig. 25-8.

Variable N, in line 110, controls the number of “wee-whoos” before the sound shuts off.

Michael L. Gatto
Los Angeles, California



Charge!

Figure 25-9 makes a very nice sound effect, using two voices to produce the rallying call so often heard at ball games.

Thomas Cechner
Naperville, Illinois

Fig. 25-9.

```

10 REM CHARGE! - THOMAS CECHNER
15 V=54272:GOSUB 75:POKE V+5,85
20 POKE V+6,85:POKE V+12,85:POKE V+13,85
25 POKE V+24,15:POKE V+4,33:POKE V+11,17
30 FOR J=1 TO 6
35 :READ H1,L1,H2,L2
40 :POKE V+1,H1:POKE V,L1
45 :POKE V+8,H2:POKE V+7,L2
50 :IF H1=50 THEN FOR T=1 TO 200:NEXT
55 :FOR T=1 TO 100:NEXT
60 NEXT J
65 GOSUB 75
70 END
75 FOR J=V TO V+23:POKE J,0:NEXT:RETURN
80 DATA 025,030,018,209,033,135,025,030
85 DATA 042,062,031,165,050,060,037,162
90 DATA 042,062,031,165,050,060,037,162
    
```



Fanfare

The routine in Fig. 25-10 produces a sound effect that adds pizzazz to title screens, game programs, and special screen announcements of all kinds. If you use it as a subroutine, you’ll have access to it whenever you like. As always, be sure that the variable names in the subroutine are different from important ones used in the body of the program.

Joseph R. Charnetski
Plains, Pennsylvania

Fig. 25-10.

```

10 REM FANFARE - JOSEPH R. CHARNETSKI
20 S=54272
30 FOR J=S TO S+23:POKE J,0:NEXT
40 POKE S+5,9:POKE S+24,15
50 FOR J=1 TO 20:READ K,L
60 POKE S+1,K:POKE S+4,33
70 FOR M=1 TO L*50:NEXT:POKE S+4,32
80 FOR N=1 TO 5:NEXT:NEXT
90 RESTORE
91 DATA 033,005,044,004,044,002,044,002
92 DATA 044,004,033,005,056,004,056,002
93 DATA 056,002,056,004,044,005,067,004
94 DATA 067,002,067,002,067,004,044,005
95 DATA 089,004,089,002,089,002,089,014

```



Bugle Calls

Some of our best-known and most rousing music can be found in military bugle calls. Musically very simple, they lend themselves perfectly to computer reproduction. Prove it to yourself by typing in the programs that follow.

The longer programs will run on either the 64 or 128. Lines 110-140 are identical in both, except for the ranges of the FOR...NEXT loops in line 150.

The shorter programs are for the 128 only. To make them easier to type and proofread, we've broken the PLAY statements into arbitrary five-character groups. Be careful—PLAY statements often use the letter O, but seldom use a zero.

Notice that while music programs are much shorter and simpler on the 128, they're still not easy for humans to read.

Assembly Figure 25-11 sounds the bugle call that's used to assemble the troops.

Taps Figure 25-12 sounds the familiar bugle call that is used at the end of the day and at military funerals.

128 Assembly If you command both troops and a 128, you can run Fig. 25-13 on the latter to gather in the former.

128 Taps Run Fig. 25-14 when it's bedtime for the soldiers.



Lullaby

The program in Fig. 25-15 plays a well-known lullaby by Johannes Brahms.


```

100 REM ASSEMBLY - LOUIS F. SANDER
110 S=54272:POKE S+5,9:POKE S+6,144:POKE S+24,15
120 FOR N=1 TO 4:READ N(N,1),N(N,2):NEXT
130 READ NO,DU:IF NO=0 THEN END
140 POKE S+1,N(NO,1):POKE S,N(NO,2):POKE S+4,17
150 FOR J=1 TO 15*DU:NEXT:POKE S+4,16
Fig. 25-11. 160 FOR J=1 TO 50:NEXT:GOTO 130
170 DATA 012,143,016,195,021,031,025,030
180 DATA 1,06,1,02,2,06,1,02,2,06,3,02
190 DATA 2,12,2,04,2,02,3,06,2,02,3,06
200 DATA 4,02,3,12,2,06,3,02,4,12,3,06
210 DATA 2,02,1,12,1,06,1,02,2,12,2,06
220 DATA 2,02,2,18,0,00

```

```

100 REM TAPS - LOUIS F. SANDER
110 S=54272:POKE S+5,9:POKE S+6,144:POKE S+24,15
120 FOR N=1 TO 4:READ N(N,1),N(N,2):NEXT
130 READ NO,DU:IF NO=0 THEN END
140 POKE S+1,N(NO,1):POKE S,N(NO,2):POKE S+4,17
150 FOR J=1 TO 50*DU:NEXT:POKE S+4,16
Fig. 25-12. 160 FOR J=1 TO 50:NEXT:GOTO 130
170 DATA 012,143,016,195,021,031,025,030
180 DATA 1,06,1,02,2,25,1,06,2,02,3,25
190 DATA 1,04,2,04,3,08,1,04,2,04,3,08
200 DATA 1,04,2,04,3,25,2,06,3,02,4,16
210 DATA 3,08,2,08,1,25,1,06,1,02,2,24
220 DATA 0,0

```

```

10 REM 128 ASSEMBLY - LOUIS F. SANDER
20 TEMPO 15
30 PLAY"T7I.O 3GSGI .O4CS O3GI. O4CSE"
Fig. 25-13. 40 PLAY"QCICS CI.ES CI.ES QOEI. CSEQG"
50 PLAY"I.ESC QO3GI .GSGQ O4CI. CSCQ."
60 PLAY"CIIII"

```

```

10 REM 128 TAPS - LOUIS F. SANDER
20 TEMPO 10
30 PLAY "T7I.O 3GSGH .O4CI .O3GS O4CH.
Fig. 25-14. 40 PLAY ".EIO3 GIO4C QEIO3 GIO4C QEIO3
50 PLAY "GIO4C H..EI .CSEH GQEQC H.O3G
60 PLAY "I.GSG H..O4 CIIII"

```

```

10 REM LULLABY - JOSEPH R.CHARNETSKI
20 R=54272:FOR J=R TO R+23:POKE J,0:NEXT
30 POKE R+5,10:POKE R+6,9:POKE R+24,15
40 FOR J=1 TO 51:READ P,T:CK=CK+P+T
50 POKE R+1,P:POKE R+4,33
60 FOR S=1 TO T*125:NEXT:POKE R+4,32
70 FOR D=1 TO 64:NEXT:NEXT:POKE R+24,0
80 IF CK<>1976 THEN PRINT"DATA ERROR"
81 DATA 31,2,31,2,37,8,31,2,31,2
82 DATA 37,8,31,2,37,2,50,4,47,6
83 DATA 42,2,42,4,37,6,28,2,31,2
84 DATA 33,8,28,2,31,2,33,8,28,2
85 DATA 33,2,47,4,42,2,37,4,47,2
86 DATA 50,8,25,2,25,2,50,8,42,2
87 DATA 33,2,37,7,31,2,25,2,33,2
88 DATA 37,2,42,4,31,2,37,8,25,2
89 DATA 25,2,50,8,42,2,33,2,37,8
90 DATA 31,2,25,2,33,4,31,6,28,2
91 DATA 25,8

```

Fig. 25-15.

The T*125 in line 60 controls the tempo at which the lullaby is played. If you run the program on a 128, you may want to change it to T*70.

*Joseph R. Charnetski
Plains, Pennsylvania*



128 Happy Birthday

If you enter the program in Fig. 25-16, you'll get a nice rendition of Happy Birthday. By adding these two lines, you'll hear it on ten different instruments.

```

5 FOR J=0 TO 9 : PLAY "T" + STR$(J)
15 SLEEP 2 : NEXT J

```

*Alejandro Ozerkovsky Adler
Mexico City, Mexico*

```

10 TEMPO 25:FOR K=1 TO 2:PLAY"MO4V1QGGHAGO5CO4WBQGGHAGO5HDWCO4QGGO5
HGEQCCO4HBAO5QFFHECDWC":NEXT:REM 128 HAPPY BIRTHDAY - OZERKOVSKY

```

Fig. 25-16.



128 Star Wars Theme

The program in Fig. 25-17 plays several bars of this well-known theme. When typing it in, the similarities in several of the lines let you

use your screen editing skills to save typing time. Lines 30 and 40 are identical, so after you enter line 30, move your cursor up, change the line number and press Return. Do a LIST and observe that both lines are present in memory.

The first parts of lines 50, 60 and 70 are also identical. Type line 50 up to the place where they differ, then use the same screen editing technique to duplicate the partial line into lines 60 and 70. Add the tail ends to lines 50-70, and your typing job is finished.

Examining this short song can reveal some of the workings of the PLAY statement. The T7 in line 20 determines what "instrument" will be playing the song. Try changing the 7 to other numbers.

Inside the PLAY statements in lines 30 through 70, notice the many places where O and a number are combined. The O is the letter O, standing for Octave. Don't make the mistake of typing a zero instead.

```

10 REM 128 STAR WARS THEME - B. KOLLAR
20 TEMPO 20 : PLAY "T7"
30 PLAY "O3 WC G IF E D O4 WC O3 HG IF E D O4 WC O3 HG IF E F WD R"
40 PLAY "O3 WC G IF E D O4 WC O3 HG IF E D O4 WC O3 HG IF E F WD R"
50 PLAY "O2 HG WA QA O3 F E D C IC D E D R O2 A HB"
60 PLAY "O2 HG WA QA O3 F E D C WG D"
70 PLAY "O2 HG WA QA O3 F E D C IC D E D R O2 A B R R O3 HG O4 QC
    O3 B A R G F E R HC WG R"

```

Fig. 25-17.

Also notice the W, H, Q and I symbols for whole, half, quarter and eighth notes respectively. Once you set one of these note lengths, all subsequent notes are played at the same length until you enter another length. In line 30, the "WC G IF E D" gives you whole notes for C and G, and eighth notes for F, E and D.

Bryan Kollar
Nanticoke, Pennsylvania



64 Keybeeper

The routine in Fig. 25-18 makes a beep each time you press a key. It has different tones for the Return key, Cursor and Function keys, and the other alphanumeric keys.

Pablo Trincavelli
Rosario, Argentina

Chapter 25

```
10 PRINT"[CLEAR,RVS,SPACE4]64 KEYBEEPER - PABLO TRINCAVELLI[SPACE4]"
20 PRINT"[DOWN]MAKES A TONE WHEN ANY 64 KEY IS PRESSED."
30 PRINT"[DOWN]STOP/RESTORE DISABLES THE FEATURE,"
40 PRINT"SYS 50000 REENABLES IT."
50 POKE 54275,8:POKE 54277,36:POKE 54278,9:POKE 254,65
60 FOR J=0 TO 54:READ K:POKE 50000+J,K:CS=CS+K:NEXT
70 IF CS=6539 THEN SYS 50000:END
80 PRINT"DATA ERROR!":STOP
90 DATA 120,169,098,141,020,003,169,195
91 DATA 141,021,003,088,169,015,141,024
92 DATA 212,096,165,197,164,254,136,201
93 DATA 064,240,022,162,128,201,001,240
94 DATA 012,162,064,201,008,144,006,201
95 DATA 051,240,002,162,016,142,001,212
96 DATA 200,140,004,212,076,049,234
```

Fig. 25-18.

☆ 128 Easy Player

Figure 25-19 is a short but powerful program that lets you explore BASIC 7.0's PLAY statement. Run it, then start pressing keys. If the PLAY statement can interpret a key, the program will make it do so; if your key is illegal in a PLAY statement, the program will give you an error message. Press A, and an A will sound. Press T then 3, and subsequent notes will sound in the drumlike T3 envelope. It works on everything but sharps and flats, and it's lots of fun to use.

Want to learn about computer music? Open your System Guide to the description of the PLAY statement, then use the program to check out the different elements. Before long, you'll feel like a byte-sized Beethoven.

```
10 PRINT"[CLEAR,RVS,SPACE4]128 EASY PLAYER - LOUIS F. SANDER[SPACE3]"
"
20 :
30 PRINT"[DOWN]PRESS KEYS FOR ANY VALID PLAY STATEMENT,"
40 PRINT"AND THEY WILL BE 'PLAYED'."
50 PRINT"[DOWN]PRESS STOP TO QUIT.[DOWN]"
60 :
70 TRAP 80:GET KEY A$:PLAY"M"+A$:GOTO 70
80 IF ER=30 THEN END:ELSE PRINT"BAD KEY!":RESUME 70
```

Fig. 25-19.

△☆ ADSR Table Maker

When using sound on Commodore computers, you always need to choose values for ADSR (Attack, Decay, Sustain and Release), then set

them by using the appropriate POKEs or other means. When doing this, it's very helpful to have access to a table that shows the Attack, Decay and Release times that correspond to the various settings between 0 and 15. (The Sustain setting is a volume, not a time).

If you have a printer, the program in Fig. 25-20 will make such a table for you. When using it, remember that a second contains 1000 milliseconds. Ten milliseconds equal one one-hundredth of a second, a hundred milliseconds equal one tenth of a second, and so forth.

```

100 PRINT"[CLEAR,RVS,SPACE7]ADSR TABLE MAKER - SANDER[SPACE8]"
110 PRINT"[DOWN]THIS PRINTS A TABLE ON THE PRINTER."
120 PRINT"[DOWN]PRESS 'P' TO PRINT, OR PRESS ANY OTHER"
130 PRINT"KEY TO QUIT."
140 GET A$:IF A$="" THEN 140
150 IF A$<>"P" THEN END
160 FOR J=1 TO 16:READ K:CS=CS+K:NEXT:RESTORE
170 IF CS<>18942 THEN PRINT"[DOWN]DATA ERROR":STOP
180 OPEN 4,4:CMD 4:PRINT" ADSR DURATIONS IN MILLISECONDS":PRINT
190 PRINT"VALUE[SPACE4]ATTACK[SPACE2]DECAY[SPACE2]RELEASE"
200 FOR V=0 TO 15:PRINT V;"[SPACE3]";:IF V<10 THEN PRINT" ";
210 READ T:FOR N=0 TO 2:D=T:IF N THEN D=3*T
220 D$=RIGHT$("[SPACE4]" +STR$(D),6)+"[SPACE2]"
230 PRINT D$;:NEXT:PRINT:NEXT:PRINT#4:CLOSE 4
240 DATA 2,8,16,24,38,56,68,80,100,250,500,800,1000,3000,5000,8000

```

Fig. 25-20.



128 TEMPO Table Maker

When working with music in BASIC 7.0, the TEMPO statement is used to control how fast the notes are played. Figure 25-21 prints a table showing the actual playing times for whole notes through sixteenth notes, for TEMPOs from 1 to 60. The printout takes 60 lines, so be sure to start with the printhead near the top of the paper.

```

100 PRINT"[CLEAR,RVS,SPACE4]128 TEMPO TABLE MAKER - CASSELL[SPACE5]"
110 PRINT"[DOWN]THIS PRINTS A TABLE ON THE PRINTER."
120 PRINT"[DOWN]PRESS 'P' TO PRINT, OR PRESS ANY OTHER"
130 PRINT"KEY TO QUIT."
140 GET A$:IF A$="" THEN 140
150 IF A$<>"P" THEN END
160 OPEN 4,4:CMD 4:PRINT,"NOTE DURATIONS IN MILLISECONDS"
170 PRINT"TEMPO[SPACE4]WHOLE[SPACE3]HALF[SPACE3]QUARTER[SPACE2]
EIGHTH[SPACE2]SIXTEENTH"
180 FOR T=1 TO 60:PRINT T;"[SPACE3]";:IF T<10 THEN PRINT" ";
190 U=19220/T:FOR N=0 TO 4:D=INT(U/2^N)
200 D$=RIGHT$("[SPACE4]" +STR$(D),6)+"[SPACE2]"
210 PRINT D$;:NEXT:PRINT:NEXT:PRINT#4:CLOSE 4

```

Fig. 25-21.

If you wonder about faster tempos, just make the appropriate change in line 180. The values of T following the FOR statement control the range of tempos printed out.

Judith Sander Cassell
Glendale, California



Musical Staff Display

Figure 25-22 is for you dabblers in computer music. It displays a musical staff with the octave numbers for each note. The 2C, for example, indicates octave 2, note C. You can use this notation directly on the 128, or as a guide to preparing POKEs for the 64.

```

100 REM MUSICAL STAFF DISPLAY - SANDER
110 PRINT"[CLEAR,WHITE,SPACE2]MUSICAL STAFF WITH OCTAVES & NOTES"
120 PRINT"[SPACE2]BY LOUIS F. SANDER[HOME]"
130 POKE 53280,15:POKE 53281,15
140 FOR J=1 TO 2:FOR K=1 TO 5
150 PRINT"[BLACK,DOWN] [SHFT *38]":NEXT
160 PRINT"[DOWN,SPACE2]----";:IF J=2 THEN PRINT"[HOME]";
170 PRINT:NEXT:PRINT"[HOME]":FOR J=1 TO 24
180 READ N$:PRINT TAB(39-3*J-36*(J>12));N$;
:IF J=24 THEN PRINT"[HOME]";
190 PRINT:NEXT:PRINT"[HOME,DOWN4]";
200 DATA"[RED]3G",3F,3E,3D,3C
210 DATA"[BLUE]2B",2A,2G,2F,2E,2D,"2C[BLACK]"
220 DATA"[RED]1B",1A,1G,1F,1E,1D,1C
230 DATA"[BLUE]0B",0A,0G,0F,"0E[BLACK]"

```

Fig. 25-22.



Sheet Music

Not all sheet music is suitable for conversion to computer music. Two kinds that I've found eminently usable are "big note" music and "fake books." Both are generally available where pianos and organs are sold, and both contain simple versions of popular and familiar melodies.

The big note music is for beginning pianists and organists, and contains lots of additional notation which is useful for computer musicians as well. One popular series is E-Z Play Today, from Hal Leonard Publishing company. Books in the series are priced at \$3.95 and up.

Fake books contain simple versions of hundreds of popular songs. Usually they are huge, and usually they sell for over \$25.00. One fake

book and a Commodore computer should be good for many years of musical enjoyment.

*Jean Patterson
Pittsburgh, Pennsylvania*

Related Tips in Other Chapters

This tip also applies to programming music and sound:

Chapter 28 64 to 128 Translations

Many of Chapter 29's "Amusements and Delights" also include sound effect routines.

Mathematics and Programming

*“The union of the mathematician with the poet,
fervor with measure, passion with correctness,
this surely is the ideal.”*

William James
Essay
1879

This chapter begins with a healthy assortment of tricks for converting numerals from base to base, a task that even the least mathematical programmer must learn to do with accuracy.

Then it moves on to rounding, roots, factorials, permutations and logarithms. If those subjects distress your poetic soul, they shouldn't; with the power of your computer, they're very easy.



Numbers, Numerals and Bases

Everybody knows that computer math uses base 2 arithmetic, and that human math uses base 10. Many computer people are also familiar with hexadecimal math, or math using base 16.

To understand these things in any depth, you need thorough understanding of several simple concepts, which will now be thoroughly explained.

The first of these is *number*. Number is an abstract concept having nothing to do with digits, bases and the like. Your fingers and my toes have little in common except their number, which in English we call “ten”. In French, that number would be called “*dix*”, while in German it would be called “*zehn*”.

The number ten, so closely allied with your fingers, is also associated with the number of players in a basketball game—two teams of five men each, for a total of ten players. The number is the same, no matter what word might be used to express it.

Although number is an abstract concept, we’ve seen that we can discuss it concretely by assigning a spoken or written word to it.

It is also very helpful to assign *symbols* to numbers. One familiar symbol for the number of toes you have is “10”. The ancient Romans, who spoke about that number as “*decem*”, used a different symbol: “X”.

Interestingly, the English, French and German speakers all use the same symbol for the number that we call ten, in spite of their very different words for it. In fact, almost every person alive today, regardless of his spoken language, uses the symbol “10” for the number we call “ten”.

That symbol is part of the *Hindu-Arabic* system of number symbols, which has become the modern world’s standard way of expressing number in symbolic form.

Number symbols, no matter how they are written, are called *numerals*. The Roman numeral for the number ten is X, while the American/French/German schoolboy’s numeral for the same thing is 10. In general conversation, we often blur the distinction between numbers and numerals, using the word “number” to represent them both. But to help you gain understanding, the rest of this section carefully distinguishes between them.

Remember that the numeral is not the number itself, but only a symbol for the number. As you will remember, the number itself is an abstract thing—a thing that we can talk about, write about, or even express in symbols, but a thing very different from the words or symbols themselves.

When we speak of binary, decimal and hexadecimal, we aren't speaking of numbers at all, but of systems of numerals. In the binary system of numerals, there are only two symbols, 0 and 1. It's called the binary system from the Greek root *bi-*, meaning two. The decimal system of numerals, taking its name from the Greek root *deci-*, has ten different symbols, the familiar digits 0 through 9. The hexadecimal system of numerals has sixteen different symbols, consisting of the ten digits 0-9, plus the letters A, B, C, D, E and F.

When we write down the numeral for the number of your fingers, we write 10, 1010, A or X, depending on whether our numerals are from the decimal, binary, hexadecimal or Roman systems. It's a potentially confusing situation but, if you understand what's presented here, for you it won't be confusing.

The number of different symbols in a numeral system is referred to as the *base* of the system. (Because Roman numerals lack the regularity of the other systems, the Roman system really doesn't have a base.)

While decimal, binary, hex and Roman are the most common numeral systems by far, there are many others. The only one you're likely to see outside of math class is the octal system, which uses base eight, and is sometimes used in computer work.

Usually, the symbols in these systems consist of some or all of the digits 0-9, plus as many letters from A-Z as are necessary. By using some or all of those 36 symbols, we could create a numeral system with any base from 2 to 36.

Now that you understand that numbers are abstract qualities of collections of things, that numerals are symbols for different numbers, and that different numeral systems can be constructed on different bases, you're ready for further understanding.



Telling Numerals Apart

Because binary, hex and decimal numerals use many of the same symbols, it can be difficult to tell which base a numeral is expressed in.

If you see the numeral 1000, for example, how do you know whether it's decimal, binary or hex? If it's decimal, it's a thousand. If binary, it's eight. If it's hexadecimal, it's four thousand ninety-six. Those are very dissimilar numbers!

When confusion like that is possible, it's good practice always to specify the base of the numeral in question. There are a half dozen widely-used conventions for making the distinction, but only three are common in the Commodore world.

First of all, the name of the base can be spelled out next to the numeral. “1010 binary” isn’t likely to be confused with “1010 hex”.

Second, the base can be specified in a subscript: 1010_2 is binary, 1010_{16} is hexadecimal and 1010_{10} is decimal, without much chance for confusion. The subscripts are always in decimal.

The final convention uses the symbol % to designate binary, and \$ to designate hex. The absence of a symbol designates decimal. In this convention, our examples would be written as %1010 and \$1010 for binary and hex respectively, and 1010 for decimal.

Stewart Terrier
Alton, Illinois



Binary for Beginners

If you want to move past the beginner stage in computing, sooner or later you’ll have to become fluent in working with binary numerals. Such fluency is absolutely essential when working with machine language, and it’s helpful when working in advanced areas of BASIC. For a detailed explanation of binary numerals, you’ll have to go to a book—good programming books usually have a chapter or section on the subject. For useful hints on working with what you find there, just study the paragraphs below.

When I moved past the beginner stage, my first accomplishment outside the decimal world was learning to count in binary; you should learn that, too. With just a little practice, it’s easy to write all the binary numerals from 0000 (for the number Americans call “zero”) to 1111 (for the number we call “fifteen”), without referring to a book or your notes. After you’ve done that several times, you’ll be very familiar with the patterns involved in binary counting, and you’ll be able to use those patterns to count as high as you’d like.

It’s also helpful to learn how to convert binary numerals into decimal form. Just start at the right, and remember that the first place is the units place, the second is the twos place, the third is the fours place, and so on through eights, 16’s, 32’s, 64’s and 128’s. Use a calculator to get the total, and soon you’ll be a binary whiz.

You can convert decimal numerals to binary by repeatedly dividing the decimal numeral by two, taking the remainders as the bits of the binary numeral. The remainder from the first division is the least significant bit (LSB), or the rightmost bit in the binary numeral. The remainder from the last division is the most significant bit (MSB), or the leftmost bit of the binary numeral. The other remainders become bits from right to left. Here’s an example of how to convert 19 decimal to binary:

- 19/2 = 9, remainder 1 (LSB = 1)
- 9/2 = 4, remainder 1 (next bit = 1)
- 4/2 = 2, remainder 0 (next bit = 0)
- 2/2 = 1, remainder 0 (next bit = 0)
- 1/2 = 0, remainder 1 (MSB = 1)

As you can see, the binary equivalent of 19 decimal is 10011.

To avoid confusion between binary and decimal numerals, computer people often use a special notation to differentiate the two. The prefix % is often used for binary, while decimal usually has no special prefix. Therefore %10011 is instantly recognizable as the binary equivalent of nineteen, while 10011, under this convention, would be the decimal numeral for ten thousand and eleven.

To become a really advanced binary expert, you'll want to commit some binary numerals to memory. Binary 0000 is easy, of course, and 1111 isn't much harder. 1010 is easy, too, because it's equal to decimal 10. Once you've memorized the other binary numerals between 0000 and 1111, you can consider yourself an expert, and you can amaze your friends with your knowledge. You'll also be well-prepared to make the move into hexadecimal.



Binary and Hexadecimal

Many beginners know that computers "think" in terms of ones and zeroes, and that binary math is therefore important to programmers. (Actually, computers think in terms of circuits being on or off; the ones and zeroes are representations of those electrical states.)

But what about hexadecimal? Who needs the additional confusion?

Actually, hex makes things simpler, not more confusing. The great benefit of hex is that it compresses a four-bit binary numeral into a hex number with a single digit. (Strictly speaking, it's a hexit, not a digit).

Not only that, but any long binary numeral can be instantly converted to a hex numeral that's one-fourth as long. So hexadecimal is just an easy shortcut for dealing with binary.

To make the conversions, you must memorize the following table:

BINARY	HEX	BINARY	HEX	BINARY	HEX	BINARY	HEX
0000	0	0100	4	1000	8	1100	C
0001	1	0101	5	1001	9	1101	D
0010	2	0110	6	1010	A	1110	E
0011	3	0111	7	1011	B	1111	F

Once you know these four-bit conversions, converting a longer binary numeral is simple: just divide it into four-bit groups and write

the hex equivalent for each group. Here are some completed conversions; use this chapter's base conversion programs to prove that they are correct:

1000	0000			binary = 80 hex
1000	0000	0001		binary = 801 hex
0001	0010	1101	1100	binary = 12DC hex
1010	1010	1010	1010	binary = AAAA hex

A. W. Grym
London, England

○△☆ Decimal to Binary Converter

The program in Fig. 26-1 converts decimal numerals to binary. It's useful for learning about binary, because it lets you see exactly what converts into what. And after you've learned to work with binary numerals, it can make your life much easier. The actual work of conversion is done in line 230. I've used this routine many times in larger programs, and it has really come in handy.

Louis F. Sander
Pittsburgh, Pennsylvania

```

100 PRINT"[CLEAR,RVS,SPACE2]BINARY TO DECIMAL CONVERTER - SANDER
[SPACE2]"
110 PRINT"[DOWN]INPUT MUST BE ALL 1'S AND 0'S, WITH"
:PRINT"NO SPACES, PLEASE!"
120 INPUT"[DOWN,SPACE4]BINARY NUMERAL";B$
130 D=0:FOR J=1 TO LEN(B$):D=D+D-(MID$(B$,J)=>"1"):NEXT
140 PRINT "[DOWN]DECIMAL EQUIVALENT=";D

```

Fig. 26-1.

○△☆ Binary to Decimal Converter

Figure 26-2 converts binary numerals to decimal. It's the complement to the program above, and has very similar uses. Like that program, it lends itself to use in other, larger, programs. The converting is done in line 130.

```

200 PRINT"[CLEAR,RVS,SPACE2]DECIMAL TO BINARY CONVERTER - SANDER
[SPACE2]"
210 PRINT"[DOWN]INPUT MUST BE BETWEEN 0 AND 255"
220 INPUT"[DOWN,SPACE2]DECIMAL NUMERAL";D
230 B$="":FOR J=0 TO 7:B$=CHR$(D AND 1 OR 48)+B$:D=D/2:NEXT
240 PRINT"[DOWN]BINARY EQUIVALENT=";B$

```

Fig. 26-2.

○△☆ **Decimal to Hex Converter**

Figure 26-3 is a one-line routine for converting a numeral from decimal to hex. If you're using a 64, be careful when typing line 120; inserting extra spaces may put it over the 80 character maximum line length.

```
100 PRINT "[CLEAR,RVS,SPACE2]DECIMAL TO HEX CONVERTER - A.W. GRYM
[SPACE2]"
110 INPUT "[DOWN]DEC NUMERAL (0-65535)";D
120 H$="":D=D/4096:FOR J=1 TO 4:D%=D:H$=H$+CHR$(48+D%-(D%>9)*7)
:D=16*(D-D%):NEXT
130 PRINT "[DOWN,SPACE10]HEX NUMERAL: ";H$
```

Fig. 26-3.

To add the routine to programs of your own, just extract the appropriate lines.

It only works for numbers from 0-65535 decimal, or 0000-FFFF hex.

A.W. Grym
London, England

○△☆ **Hex to Decimal Converter**

The routine in Fig. 26-4 will convert a four-digit hexadecimal numeral to its decimal equivalent. Be careful when typing line 220; if you insert extra spaces, its length may exceed the 64's maximum of 80 characters.

```
200 PRINT "[CLEAR,RVS,SPACE2]HEX TO DECIMAL CONVERTER - A.W. GRYM
[SPACE2]"
210 INPUT "[DOWN]4-DIGIT HEX NUMERAL";H$
220 D=0:FOR J=1 TO 4:D%=ASC(H$):D%=D%-48+(D%>64)*7:H$=MID$(H$,2)
:D=16*D+D%:NEXT
230 PRINT "[DOWN,SPACE8]DEC NUMERAL: ";D
```

Fig. 26-4.

It requires you to enter hex numerals with four places, so use leading zeroes if you need them.

After making the appropriate modifications, you can use this routine in other programs.

A.W. Grym
London, England

○△☆ Universal Base Converter

The program in Fig. 26-5 will convert a numeral from any base to any other, if both bases are in the range two to 36 decimal.

*Art Skiles
Huntington, Indiana*

```

100 PRINT"[CLEAR,RVS,SPACE2]UNIVERSAL BASE CONVERTER - ART SKILES "
110 ER$="[^2]?":N$="":INPUT "[DOWN2]NUMERAL TO CONVERT";N$
    :IF N$="" THEN END
120 INPUT"[DOWN,SPACE2]FROM BASE (2-36)";F
130 IF F<2 OR F>36 THEN PRINT TAB(20);ER$:GOTO 120
140 INPUT"[DOWN,SPACE4]TO BASE (2-36)";T
150 IF T<2 OR T>36 THEN PRINT TAB(20);ER$:GOTO 140
160 X=0:FOR Z=1 TO LEN(N$):A=ASC(MID$(N$,Z,1))
170 B=A+48*(A>47 AND A<58)+55*(A>64 AND A<91)+85*(A>96 AND A<123)
180 IF B>=F THEN PRINT "[DOWN]ERROR IN NUMERAL TO CONVERT!":GOTO 110
190 X=X*F+B:NEXT Z:CN$=""
200 B=X-INT(X/T)*T:X=(X-B)/T
210 A=B-48*(B<10)-55*(B>9)
220 CN$=CHR$(A)+CN$:IF X THEN 200
230 PRINT "[DOWN] CONVERTED NUMERAL: ";CN$

```

Fig. 26-5.

○☆ 128 Arabic to Roman Conversion

Figure 26-6 makes those pesky Roman Numeral conversions a breeze! You can use it when you're designing the cornerstones for government buildings, composing the ending credits for motion pictures, or channeling with citizens of ancient Rome.

Actually, it's a powerful little program that accepts Arabic numerals into the millions, and converts them quickly and flawlessly. With a little additional programming, you could output to a printer, or even print all the Roman numerals from 1 to 3,999,999.

To express numbers larger than 3999, Caesar, Brutus and this program draw lines over the letters in the Roman numerals. The letter M stands for one thousand, but M with a line over it stands for a thousand thousand, or a million.

*Jim O'Neill
Whiting, New Jersey*

Chapter 26

```
10 PRINT"[CLEAR,RVS]128 ARABIC TO ROMAN CONVERSION - O'NEILL[RVOFF]"
20 T=25:DIM A$(T),A(T):R$="[UP,CMDR @,DOWN,LEFT]"
   :S$="[UP,CMDR @2,DOWN,LEFT2]":FOR D=13 TO T:READ A$(D),A(D):NEXT
   :FOR D=1 TO 12:A(D)=A(D+12)*1000:NEXT
30 FOR D=1 TO 11 STEP 2:M$=M$+"[DOWN4]":A$(D)=R$+A$(D+12)
   :A$(D+1)=S$+A$(D+13):NEXT:C$="M"+R$:M$=LEFT$(M$,21):P$="[LEFT]"
   :A$(10)=C$+"X"
40 N=3999999:PRINT"[HOME,DOWN2]INTEGER FROM 1 TO"N"(0 TO END):"
   :A$(12)=C$+"V":R$=CHR$(13)
50 K$="":J$="":WINDOW 0,3,38,3,1
60 DO:K$=K$+J$:L=LEN(K$):PRINT"[HOME]";K$;"[CMDR @]";:GET KEY J$
   :IF L>0 AND J$=CHR$(20) THEN K$=LEFT$(K$,L-1)
   :PRINT"[LEFT] [HOME]";K$;"[CMDR @]";:ELSE IF L>0 AND J$=R$ THEN
   EXIT
70 Q$="[HOME2,DOWN]":IF VAL(J$)=0 AND J$<>"0" THEN J$=""
80 LOOP WHILE L<T:PRINT P$:K=VAL(K$):IF K>N OR J$<>R$ THEN 50
   :ELSE IF K=0 THEN PRINT Q$;"[DOWN]":END
90 K$=STR$(K):FOR D=LEN(K$) TO 8:K$=" "+K$:NEXT:WINDOW 0,5,39,24
   :PRINT M$;K$" = ";
100 Q=0:DO:Q=Q+1:LOOP UNTIL K-A(Q)>=0:K=K-A(Q):PRINT A$(Q);
   :IF K=0 THEN 50:ELSE 100
110 DATA M,1000,CM,900,D,500,CD,400,C,100,XC,90,L,50,XL,40,X,10,IX,
   9,V,5,IV,4,I,1
```

Fig. 26-6.

○△ 64 Binary Key Decoder

Figure 26-7's little program lets you see the ASCII value of any key on the 64's keyboard, but it expresses the value in binary notation rather than the usual decimal or hex. Programmers who frequently use machine code may find it helpful, as will readers who want to learn more about binary. After all, the best way to learn a new numeral system is to see how it expresses some quantities with which you're already familiar.

```
10 PRINT"[CLEAR,RVS] 64 BINARY KEY DECODER - IMAD M. AHMAD[SPACE2]"
20 FOR J=0 TO 33:READ K:CS=CS+K:POKE 49152+J,K:NEXT
30 IF CS<>5366 THEN PRINT"DATA ERROR":STOP
40 PRINT"[DOWN2]PRESS ANY KEY TO SEE ITS ASCII CODE IN
50 PRINT"[DOWN]BINARY.[SPACE2]PRESS STOP TO QUIT."
60 PRINT:SYS 49152:GOTO 60
70 :
71 DATA 032,228,255,240,251,032,210,255
72 DATA 133,251,169,017,032,210,255,165
73 DATA 251,133,251,162,008,006,251,169
74 DATA 048,105,000,032,210,255,202,208
75 DATA 244,096
```

Fig. 26-7.

Imad M. Ahmad
Amman, Jordan



All About Rounding

When programming business or mathematical applications in BASIC, the subject of rounding comes up in several different ways. This tip gives methods of dealing with two of them.

First, the computer itself sometimes introduces small errors that need to be rounded off by the program. This problem occurs because the computer does its math by flipping bits in binary; to work with decimal numbers, it converts them to binary representation, processes them, then converts them back to decimal form. During the binary processing, the computer sometimes needs to do rounding or truncating of its own. When this happens and the binary is converted to decimal, the result is very slightly different than what we would expect.

To see this error creation in action, execute this line then watch your screen:

```
FOR X = 1 TO 15 : PRINT SQR(X*X) - X : NEXT
```

Mathematically, the square root of X times X is surely X. And X minus X is surely zero. But as you can see on the screen, the computer introduces some tiny errors. For more of the same, execute this:

```
100 FOR X=6 TO 12 STEP 0.1
200 PRINT X
300 NEXT
```

As the numbers flash by on the screen, you'll easily see the errors in the eighth decimal place.

One way to eliminate this type of error is to pass X through the STR\$ and VAL functions. You can see the effect by adding this to the program above:

```
110 X$=STR$(X) : X=VAL(X$)
```

The second angle on rounding comes when the logic of a program requires that a number be rounded, even though that number is perfectly accurate. For example, when a program dealing with dollars and cents encounters a result with three or more decimal places, it's usually desirable to round that result to two places.

There are many routines to do this sort of work. The one in Fig. 26-8 uses a User Defined Function, FNR, to do the rounding.

Chapter 26

```
100 PRINT "[CLEAR,RVS,SPACE5]ALL ABOUT ROUNDING - J. GREER[SPACE6]"
110 :
120 DEF FN R(N1)=INT(N1/N2+0.5)*N2
130 REM N1=NUMBER TO BE ROUNDED
140 REM N2=WHAT TO ROUND IT TO: .01 ETC
150 :
160 PRINT
170 N1=0:INPUT"[SPACE4]# TO BE ROUNDED";N1:IF N1=0 THEN END
180 INPUT"ROUND TO (.01 ETC.)";N2
190 PRINT"[SPACE3]# AFTER ROUNDING=";FN R(N1)
200 GOTO 160
```

Fig. 26-8.

The function definition is contained in line 120. It must be executed before the rounding routine will work. N1 is the number to be rounded, and N2 is the decimal representation of the closest power of ten to be rounded to. If N2 = 0.01, the function will round to the nearest hundredth. If N2 = 100, the function will round to the nearest 100.

Lines 160-200 let you test the routine for various values of N1 and N2.

Johnathan Greer
Tyrane, Pennsylvania



Roots

BASIC's SQR function will easily find the square root of any number that the computer can handle. But what if you need a cube root or other root? Just use this simple expression, where N is the number whose root you want, and R is the root you're looking for:

$$X = N^{(1/R)}$$

For example, to derive the fourth root of 32768, enter

```
PRINT 32768^(1/4)
```

Ann H. Hartmann
West View, Pennsylvania



Factorials

In mathematics, the factorial of N is defined as the product of all the integers between 1 and N. Commodore computers don't have a built-in factorial calculator, but you can easily compute them with the following routine, where N is the number whose factorial you need:

```
100 F = 1 : FOR J = 1 TO N : F = F*J : NEXT
```

After execution, F will contain the factorial of N.

A. W. Grym
London, England



Permutations and Combinations

Permutations and combinations are studied in college and in advanced high school math, and understanding them helps solve many problems in probability and statistics. Fortunately, this fascinating mathematical subject is readily amenable to computerization.

A *permutation* is an ordered arrangement of a number of symbols or objects. XYZ, ZYX and YXZ are permutations of the symbols X, Y, and Z.

A *combination* is a permutation that includes the same objects, regardless of the order in which they are arranged. XYZ, ZYX and YXZ are all different arrangements of the same combination. WXY, WXZ and XYZ are different combinations.

Baseball batting orders are good examples of permutations. When the same nine players bat in different orders in different games, we are looking at different ordered arrangements of the same nine men. In other words, we are looking at different permutations.

If you wanted to know how many different batting orders would be possible on a nine-man baseball team, you'd ask a mathematician "how many permutations are there of nine objects taken nine at a time?" If you could choose your batting order from a 24-man roster, you'd ask "how many permutations are there of 24 objects taken nine at a time?"

```

100 PRINT"[CLEAR,RVS] PERMUTATIONS AND COMBINATIONS - SANDER "
110 PRINT"[DOWN,SPACE2]THIS SHOWS THE NUMBER OF POSSIBLE"
120 PRINT"[SPACE2]PERMUTATIONS AND COMBINATIONS OF"
130 PRINT"[SPACE2]NUMBERS THAT YOU INPUT."
140 INPUT"[DOWN2,SPACE8]HOW MANY OBJECTS";N
150 INPUT"[DOWN]TAKEN HOW MANY AT A TIME";D
160 IF D>N THEN PRINT"[DOWN]TOO MANY!":GOTO 230
170 P=1:Q=1
180 FOR J=N-D+1 TO N:IF 1.7E38/J>=P THEN 200
190 PRINT"[DOWN,SPACE2]> 1.7E38 PERMUTATIONS!!":GOTO 250
200 P=P*J:NEXT:FOR K=2 TO D:Q=Q*K:NEXT
210 C=P/Q:IF D=1 THEN C=N
220 PRINT"[DOWN] NUMBER OF PERMUTATIONS =";P
230 PRINT"[DOWN] NUMBER OF COMBINATIONS =";C
240 PRINT
250 END

```

Fig. 26-9.

Now that you've seen an example of permutations, what about combinations?

Let's say you had seven basketball players in a gym, and wanted to know how many different five-man lineups you could put together. In this case it's only important which players are on the team; the order doesn't matter at all. You'd call your mathematician and ask "how many combinations are there of seven things taken five at a time?"

The simple program in Fig. 26-9 can stand in for your mathematician, and can give answers a lot faster than he can. Just enter the number of objects (or numbers, or players, or whatever), and the number to take at a time. The answer will come up in milliseconds.



Logarithms

As every well-taught high school student knows, logarithms are exponents. Most of us who know about them are accustomed to *common logarithms*, or logs to the base 10. But BASIC's logs are *natural logarithms*, or logs to the base e (2.71828). Natural logs are widely used in electrical engineering and other fields.

If you want to work with common logs, you can define a function that will make the conversion for you, as follows:

```
10 DEF FNL(N) = LOG(N)/LOG(10)
```

Once line 10 has been executed, FNL(q) will give the common log of q , where q is any numeric constant or expression. Calculating FNL(10000) will return a value of 4, which of course is the common log of 10000.

EXP, a BASIC function closely associated with LOG, calculates the value of e raised to the power inside the parentheses following the EXP. If you need a similar function using base 10 rather than e , you can use:

```
20 DEF FNE(N) = EXP(N*LOG(10))
```

You can test it by getting FNE(4), or 10 raised to the fourth power. If you know your high school math, you know that that's 10000.

Ann H. Hartmann
West View, Pennsylvania



Working with Large Numbers

The largest number that BASIC can represent is approximately 1.7 times 10 to the 38th power, or 1.7E + 38 in exponential notation. (This is the number 17 followed by 37 zeroes.) While that is a very large number, on occasion it's too small for your needs. (If a calculation attempts to exceed the limit, an Overflow Error results.)

If you use a few programming tricks, you can use logarithms to handle numbers larger than BASIC's limit. When using logarithms, multiplication becomes addition, division becomes subtraction, and exponentiation becomes multiplication. For example:

$$\begin{aligned}\text{LOG}(X*Y) &= \text{LOG}(X) + \text{LOG}(Y) \\ \text{LOG}(X/Y) &= \text{LOG}(X) - \text{LOG}(Y) \\ \text{LOG}(X^Y) &= \text{LOG}(X) * Y\end{aligned}$$

The program in Fig. 26-10 illustrates how logs can be used in calculating factorials. If you calculate factorials normally, you will get an overflow for anything higher than the factorial of 33. With our program, you can go a lot higher than that. For example, running it for $N = 100$ gives $N! = 9.3326192 \text{ E } 157$.

A.W. Grym
London, England

```
10 REM WORKING WITH LARGE NUMBERS
20 PRINT "[CLEAR]FACTORIAL FINDER - A.W.GRYM"
30 PRINT "[DOWN]ILLUSTRATES USE OF LOGS. [DOWN]"
40 INPUT "N";N
50 F=LOG(1):FOR Z=1 TO N:F=F+LOG(Z):NEXT
60 F=F/LOG(10):E=INT(F):M=10^(F-INT(F))
70 PRINT "N!=";M;"E";E
```

Fig. 26-10.

Related Tips in Other Chapters

If you're mathematically inclined, you'll enjoy these additional tips:

Chapter 5	Typing Large Numbers Abbreviating Zero
Chapter 19	INTEResting Bug
Chapter 23	Bytes
Chapter 30	Lottery Assistant Dice Machine

Debugging

*“A little, wretched despicable creature;
a worm, a mere nothing, and less than nothing;
a vile insect that has risen up in contempt
against the majesty of Heaven and earth.”*

Jonathan Edwards
The Justice of God in the Damnation of Sinners
1734

Invidious, insidious insectile invaders. Bugs. Tiny, troublesome, torturing, tenacious — trapping them takes tremendous time. Sometimes they're so serious we throw our work away.

These tips will help you find those bugs. These tips will help you squash them.



Easier Bug-spotting

Sometimes the hardest part of debugging is finding the offending bit of code in among all the good stuff. If you list your program on the printer, you'll be surprised how the bugs become more visible.

Another debugging secret is to simultaneously press the Shift and Commodore keys to put the computer into lowercase mode. If you've managed to use the letter O where a zero was required, or if you've substituted a one for an I, the lowercase display will make your error more obvious.

When you're looking for a bug in an on-screen listing, it's usually helpful to change your cursor color to something different than the color of the letters. Then move the cursor, character by character, over your listing, and keep your eyes out for errors.

*W. A. Sander
Charleston, South Carolina*



Let the Computer Read It

Often it is difficult to find errors in listings and in DATA statements. But if you have a voice synthesizer, you can get it to read the listing to you at whatever speed you desire.

Like all good computer equipment, it never needs to be bribed or persuaded to help you, and it never gets too tired or too busy to work.

*Perry Wilson
Des Moines, Iowa*



Finding Syntax Errors

When you're stumped by a Syntax Error message in a line that looks error free, try the following:

- 1) Change character sets by simultaneously pressing the Shift and Commodore keys. This finds many hard-to-see typing mistakes, such as the letter O entered instead of zero or Shift L instead of L.
- 2) Insert spaces between keywords and variables. This not only makes the line easier for you to proofread, but it also makes it easier for the computer to interpret. The computer thinks TANDU means TAN DU, even if you think it means T AND U.
- 3) Look for reserved variables used incorrectly; also look for keywords used as variables. S = ST is fine, but ST = S causes an error; FB = X works, but FN = X does not.

4) Get a second opinion. Compare your syntax against examples in the manual, or ask a friend to look at the problem line.

*Nicholas G. D. Old
Baltimore, Maryland*



Deceptive Lines

If you're debugging a program and you encounter Syntax Errors in a line that seems all right, try listing the line by itself. You might just get what looks like two lines instead of one. If you do, you've found the bug.

What you're seeing is really a flawed single line produced by a keyboarding error. These lines are caused by typing a line exactly as wide as the screen, then typing another without pressing Return. To your careless eyes, you've entered two lines. But to the computer, you've only entered one, and it has a line number somewhere in its middle. That line number is what causes the Syntax Error.

You can eliminate the bug by typing both lines again.

*Anna M. A. Busch
Charleston, South Carolina*



Unseen Characters in a String

Have you ever had a string print out strangely on the screen, or even clear the screen entirely? If so, it was probably because there were some control characters embedded in the string.

One way to test for this is to use the LEN function to find the length of the string, then to compare this length with the visible portion of the string after printing. For example, if the string is A\$, then you might:

```
PRINT LEN(A$) : PRINT A$
```

Another trick is to print a quotation mark before the string. This lets you see the quote-mode equivalents of the control characters. In our example you could do this by:

```
PRINT LEN(A$) : PRINT CHR$(34);A$
```

Sometimes it's helpful to see exactly where the string begins and ends on the screen. For this, I use the quote-pound-quote sequence: "#". The quotes and the "pound sign" are next to one another on the keyboard, so it's very easy to type the sequence. An example of using this would be:

```
PRINT "# " A$ "#"
```


Please note from our examples that the semicolon is optional as a separator of the items you are printing. I usually like to use it, however, because it makes programs easier to read.

Mike Reeves
Klamath Falls, Oregon



Debugger's Nightmares

Some of the worst debugging problems occur when the computer reports an error in a line other than the line that contains the error. Several conditions can cause that situation, and unless you're aware of them, you could spend lots of time and energy looking for a mistake in a perfect line.

Usually these problems occur in READ...DATA statements or in user-defined functions (DEF FN). Here are two examples of what can happen; study them now, and you might remember them when the debugger's nightmare happens to you.

First, consider these two lines, which could be two isolated lines in a larger program of some sort:

```
100 FOR J=1 TO 3 : READ K : NEXT
500 DATA 10,20,OOPS
```

Line 100 is looking for three numeric DATA items, but the third DATA item is a string. When the program attempts to read that item, the following error will occur:

?TYPE MISMATCH ERROR IN 100

Unless the programmer is in the know, he or she can waste a lot of energy examining line 100, which doesn't have an error at all. Even if you are in the know, you can waste a lot of time looking for the offending DATA statement, which the computer does not identify for you. This chapter's tip on Debugging DATA statements can help you when this happens.

Here's another nightmare:

```
100 DEF FNA(K)=NEXT
800 B=FNA(K)
```

There is an error in line 100, since NEXT is illegal as a definition for a function. But when line 100 is executed, the computer will not detect the error! That won't happen until line 800 is executed, at which time the computer will report:

?SYNTAX ERROR IN 800

Much as with the READ...DATA error, the problem is in a line referred to by the reported line, rather than in the reported line itself. Pity the man who loses sleep while trying to correct line 800.

Well there they are, dear readers—two nightmares that can keep you awake all night. Be aware of them, and hope that you remember this tip when they come into your life.



Debugger's Helper

When a program isn't running the way it should, sometimes it's questionable whether a certain line was actually executed. A very simple way to find out is to insert an illegal character such as a bracket immediately after the line number in the line you want to check. Then run the program again, and if it tries to execute your line, the computer will stop with a Syntax Error. Delete the illegal character to return your program to normal. If you don't get the error, of course, it's positive proof that the line in question was not executed.

*Kapil Raina
Neshanic Station, New Jersey*



Debugging Using STOP and CONT

These two statements were created as debugging aids, but it's surprising how many people don't know how to use them.

When the computer encounters a STOP command, program execution ceases and a message appears on the screen. The message is BREAK IN LINE XXX, where XXX is the number of the line in which the STOP was found. After the error message, of course, you'll see the READY prompt and a flashing cursor.

The neat part of all this is that program execution can be resumed by typing CONT and pressing Return. The computer will begin by executing the statement immediately after the STOP, and all variables will contain their previously existing values.

So if you want to know whether a certain part of your program is being executed, temporarily put a STOP statement in it somewhere. If you get the BREAK message, you'll know that its own line, at least, is being processed. Then to continue just enter a direct mode CONT.

In a complicated debugging situation, you can embed several STOPS in the program. Like the obedient servants they are, each one will identify itself by line number, and you can use CONTs to step right through them.

You can even use STOP in debugging a multi-statement line where you can't find the source of the error. Put a STOP after the first colon

in the line, then run the program. If it reaches the STOP, you know that everything up to that point is O.K. Now list your line, delete the STOP, and insert another STOP after the second colon in the line. Repeat the process until you get your Syntax Error instead of a Break. When that happens, you know that the error is someplace between the STOP's previous and present locations.

*Orkin Rollins
Hemiptera Lake, Florida*



Dump Those Variables

When debugging a program, it's often helpful to know the value each variable had when the program stopped running. Amazingly, all you need to do is ask! When a program stops due to a crash or a normal termination, the computer remembers the values of every variable the program has set. You can see the values yourself by executing the appropriate direct mode statements. To see the value of F, just enter PRINT F in direct mode. To see KM\$, enter PRINT KM\$. If you want to look at an entire array, say L(S), do something like this:

```
FOR S = 0 TO 10 : PRINT S,L(S) : NEXT
```

As long as you are careful, you can explore the variables as thoroughly as you'd like. The care is required because certain normally innocuous actions can reset all the variables to zero. On the 64, for example, if you make any changes to the program, all the variables are reset. Such changes include adding a line, deleting a line, or even changing a line. When you do any one of them, it's goodbye time for your variables.

Any 128 owners are lucky here, because BASIC 7.0 retains all variable values even when lines are changed. That feature makes for some delightfully easy debugging, if such a thing can exist. But be careful here as well: if you're working in 64 mode, this feature isn't present.

If you need to check a lot of variables, you can automate the process by adding some lines like these to your program:

```
49999 END
50000 PRINT "KM=" ; KM
50010 PRINT "ML=" ; ML
50020 FOR J=1 TO 7 : PRINT L(J) : NEXT
50030 STOP
```

When your program stops and you need to dump these variables, just do a direct mode GOTO 50000. (GOTO doesn't reset your variables; RUN does.)

I've used these tricks many times over the years, and they've saved me countless hours of debugging time. I hope they work as well for you.

*Christopher Mead
Athens, Ohio*



128 Variables

One very nice feature of the 128 is that variables are stored in their own bank of memory, meaning that they retain their values even after a BASIC line has been entered, deleted or changed. You can fix a bad line, then use GOTO to resume your program, with all variables and arrays intact.

Program chaining now works reliably, as well—you no longer have to be sure that the first program is longer than all the subsequent ones, and you can use static strings like G\$ = "GOLDMINE", because these too are stored in the variable bank.

*Mike Tranchemontagne
Nashua, New Hampshire*



128 Editing Precaution

An advantage of the 128 is that BASIC retains its variables when the program is changed. This lets you stop your program, edit it, then continue where you left off with the GOTO command. Needless to say, this is impossible with the 64 and most other computers.

There's a downside to this advantage, however. If you are in a subroutine and do this editing in the main program, the location of the GOSUB command may get changed. When your restarted program attempts to execute the inevitable RETURN, it will attempt to return to the wrong place, with unpredictable results. So use this feature with care.

*Ian Adam
Vancouver, British Columbia
Canada*



Debugging DATA Statements

Many BASIC programs contain line after line of DATA statements where the data items are numbers between 0 and 255. Those numbers are actually the steps in a machine language program that the BASIC program is putting into memory by use of POKE statements. After the machine language has been POKEd, it's activated by a SYS statement, which is the machine language equivalent of RUN.

Because the DATA numbers are the decimal version of a machine language program, they don't make any sense to the human beings who type them. And because they don't make any sense of their own, it's very easy to make mistakes when typing them. Such mistakes can be maddeningly difficult to find, as you know if you've ever made one. Here are some tips to help you find them:

- If the program includes the four-letter checksums from Commodore's Magazine Entry Programs—use them. The Magazine Entry Programs are designed to detect typing errors, and are described in detail in every issue of this magazine. They are well worth the effort of typing them in.

- When there is an error in a DATA statement, the computer often reports it as occurring in the line that READs the faulty statement, rather than in the DATA line itself. For debugging these problems, you can add other statements to the READ section of your BASIC program. The easiest to add is a PRINT statement, which will put each data item on the screen as it is POKEd, allowing you to observe the process as it occurs. If a typing error has put one of your DATA items outside the range of 0-255, you'll get an Illegal Quantity Error when the program tries to POKE it. Careful use of the PRINT statement can lead you directly to the error. For example, if your program has a line like this:

```
170 READ K : POKE J,K
```

Alter it temporarily to this:

```
170 READ K : PRINT K : POKE J,K
```

The new statement PRINTs the DATA item before the POKE is tried. If you get an Illegal Quantity Error, for example, the most recently printed DATA item is the culprit. Because you see it on the screen in context with its predecessors, you should be easily able to find it in the program.

- You can take this idea one step further by printing not only the DATA item, but also the line number in which it appears. On the 64 and Plus/4, that line number is found in memory locations 63 and 64. On the 128, it is found in locations 65 and 66. The line number can be printed using something like this:

```
170 READ K : PRINT K, PEEK(63) + 256*PEEK(64) : POKE J,K
```

For the 128, just PEEK 65 and 66, rather than 63 and 64.

- If the data items and line numbers print too quickly to the screen, you can use the Control or Commodore key to slow them down. (The key you use depends on whether you have a 64 or a 128.) To take the slowdown one step further, add this line:

```
171 GET A$ : IF A$ = " " THEN 171
```

This will look for a keypress after each data item is READ, PRINTed and POKEd. To move ahead quickly, just hold down the SPACE bar (it repeats!). To move step-by-step, use any non-repeating key.

- If you still have problems after using the above steps, you should carefully proofread your program, bearing in mind that the printed program is almost never in error. When proofreading, it's helpful to involve someone other than yourself, because you will be somewhat distressed, and because you have already overlooked the error several times. Common mistakes to look for at this stage are: leaving out a DATA item or a whole line; duplicating an item or a line; putting extra commas inside or at the end of a line; and using a period instead of a comma. I'm sure your creative mind can conjure up some others, but these are the ones most often seen.



TAB Errors

If you get mysterious ?Bad Subscript errors in a TAB statement, you have probably put a space between the TAB and the first parenthesis that follows it. The space makes the computer think you want TAB to be the name of an array; if the number in parentheses is greater than eleven, you'll get the error.

*Taryn Puleo
New Paltz, New York*

Related Tips in Other Chapters

These tips will also help in debugging:

Chapter 2	Typing in Programs
	Read It to the End
Chapter 19	INTeresting Bug
Chapter 23	Editing SEQ Files
Chapter 32	64 Variable Dump

Translating

*“Bless thee, Bottom! bless thee!
thou art translated.”*

William Shakespeare
A Midsummer-Night's Dream
1595-1596

One of Commodore's truly unique accomplishments was making the new 128 perfectly emulate the old 64. In spite of this world-class compatibility, there are times when we need to translate a program so it will run in a different mode, or on a different machine. Here are a few ways of accomplishing the translations.

☆ **64 to 128 Translations**

Many BASIC programs written for the 64 will run as is on the 128 in 128 mode. Sound and screen POKEs are in the same locations in both machines. Only programs with sprites, machine language, or lots of PEEKs and POKEs will not run on both machines. So if you don't know whether a program will run in 128 mode, give it a try. You have everything to gain and nothing to lose.

*Jay Machielse
Grand Blanc, Michigan*

△ **128 Programs and the 64 Super Expander**

Commodore's Super Expander 64 cartridge is available many places for about \$20, and adds many music and graphics commands to the 64. With minor modifications to the programs, you can run many 128 music and graphics programs on the 64 with Super Expander, including many of those found in "Lou Sander's Tips and Tricks"!

*Michael Hall
DeSoto, Texas*

△☆ **128 Sprites on the 64**

Here is a procedure that will let you use the 128's powerful SPRDEF system to design sprites for the 64.

First enter the sprite editor using the SPRDEF command, and respond to the sprite number prompt with a 1. Next design your sprite and save it using Shifted Return. Then press Return to exit the sprite editor.

Type MONITOR and press Return, and when you get the Monitor prompt, enter this command:

```
T F0E00 F0E40 0C000
```

While still in the monitor, enter:

```
G FFF4D
```

this will put you in 64 mode, with your sprite in memory locations 49152-49215. You can use PEEKs and POKEs to move it to the cassette buffer or other suitable location.

*Robert Elzinga
Fairport, New York*

△☆ 64 Sprites on the 128

It's easy to convert sprite programs for the 64 to work in 128 mode, if you know a few secrets. Here they are:

1) 64 sprites can be located anywhere in memory, while 128 sprites have a special area reserved for them at 3584-4095. Because you'll probably never find a 64 program that locates them in 3584-4095, your converted program will have to put them there itself.

2) The 64 sprite locations often contain all zeroes at power up, while the 128 sprite locations never do. On the 128, then, your program should zero out these locations before POKING the sprites.

3) In both computers, the "sprite pointers," locations 2040-2047, tell the computer where the sprites are located in memory. The values in the pointers must equal the address of the first byte of the corresponding sprite, divided by 64.

4) In the 128, POKES to the sprite location registers won't work unless you first POKE 4861 with a non-zero value. This all-important POKE disables the MOVSPR command, which otherwise controls the positions of the sprites. It's a good practice to POKE 53264,0 at the same time; this puts all the sprites onto the left side of the screen, where most 64 programs assume them to be at power up. If you fail to make this POKE, it's possible that MOVSPR has put them on the right where you don't want them.

The POKE to 4861 also disables the PLAY statement and some other features of BASIC 7.0, so you shouldn't add BASIC 7.0 statements to your converted program.

You can reenable MOVSPR and the rest of BASIC 7.0 by simultaneously pressing Stop and Restore, or by returning POKE 4861,0.

5) Your program can detect whether it's running in 64 mode or 128 mode by PEEKing location 65534. If the result is 23, the system is in 128 mode.

I use the following lines of code to let my sprites work in either mode. They assume that the sprites are put into 64 memory starting at location 832; if your program puts them somewhere else, substitute that location for the 832 in line 10:

```

10 SL=832:IF PEEK(65534)=23 THEN SL=3584:POKE 4861,1:POKE 53264,0
20 FOR J=0 TO 511:POKE SL+J,0:NEXT
30 FOR J=0 TO 7:POKE 2040+J,J+SL/64:NEXT
40 This line should POKE your sprites into memory at SL upward.

```

Once these lines are in the program, I look through the rest of it for POKEs to 2040-2047, changing them as required by the new situation.

To reset the 128 to normal operation when the program is finished, I include something like this at the end:

```
50000 IF PEEK(65534)=23 THEN POKE 4861,0
50010 END
```

Now that you know the secrets, you should practice them by converting a program or two. The "64 Paratrooper" program in Chapter 29 would be an interesting one to start with.



64 to PET Translation

Many schools are still using Commodore PET computers which, unfortunately, cannot directly load programs written on the 64. The BASIC on the two machines is very similar, so many 64 programs can run on the PET, if you can get them to load. If you're willing to do some finagling when saving or loading, your PET will readily digest what the 64 turns out.

If a program has been saved in the conventional way on the 64, you can load it into the PET by following these steps:

- 1) Put the disk in the PET drive.
- 2) Type the following, ending each item with a Return:

```
0 REM
POKE 2048,0
```

- 3) Load the program as you normally do.
- 4) Type these lines, ending each one with a Return:

```
POKE 1025,1
POKE 1026,8
```

- 5) Delete line 0.
- 6) Save the program onto a different disk.

The program you have saved can now be loaded into the PET with no special steps required.

If you want to save a program from your 64 so it can be loaded directly into the PET, do this:

- 1) Being careful not to let the line exceed 80 characters, type this on one continuous line, ended with Return:

POKE56576,5 : POKE53272,4 : POKE648,128 :
POKE 1024,0 :POKE44,4 : POKE56,128 : NEW

- 2) Clear the screen.
- 3) Load your program into the 64.
- 4) Save the program onto a different disk.

The program you have saved can be loaded directly into the PET.

*Arthur A. Abramovitch
Gloucester, Ontario
Canada*

Related Tips in Other Chapters

This tip may also be useful in making program translations:

Chapter 23 Machine Type Detector

Amusements and Delights

*“I am a great friend to public amusements;
for they keep people from vice.”*

Samuel Johnson
Letter
March 31, 1772

As every schoolboy knows, the personal computer is much more than man's greatest intellectual tool: it's his most awesome source of addicting entertainment.

If your schooldays predate the personal computer age, the type-in programs in this chapter will bring you up to date. If you're a scholar of the 80's, you'll find them totally rad.

○△☆ Mind Reader

Your computer has a mind of its own. To see its innermost thoughts, run the program in Fig. 29-1.

```
100 REM MIND READER - LOUIS F. SANDER
110 :
120 REM SHOWS WHAT IS *REALLY*
130 REM ON YOUR COMPUTER'S MIND
140 :
150 PRINT CHR$(147);TAB(209);TAB(211);:FOR J=16 TO 1 STEP-1
160 PRINT CHR$(ASC(MID$("REQYL$WM$VVI$SX$",J,1))-4);
170 NEXT:PRINT TAB(253);TAB(249)
```

Fig. 29-1.

○△☆ Burrow

In 1978, when you could buy an 8K Commodore PET for \$795, Fig. 29-2's active and amusing program appeared in the "One-Liners" section of The PET Gazette, one of the earliest newsletters for Commodore owners. Over the years it has been used and reprinted by hundreds of user groups. Type it in and try it. The little critter has not grown stale with age.

```
0 REM BURROW - THE PET GAZETTE
1 A$="[UP,DOWN,LEFT,RIGHT]":PRINT MID$(A$,RND(.5)*4+1,1);"*[LEFT]";
:FOR J=1 TO 30:NEXT:PRINT"[RVS] [LEFT]";:GOTO 1
```

Fig. 29-2.

We've spread out the listing so you can see what to type, but it's essential that you do not type any spaces between the words in the line. If you attempt to include them, the line will exceed 80 characters and cannot be entered into a 64. The ONLY space in the line is the one in the PRINT statement, between the [RVS] and the [LEFT].

△☆ Potholes

In Fig. 29-3's short but very exciting program, you're an unfortunate driver who's stumbled onto a highway awaiting major repairs. Being the wild and crazy guy that you are, you floor your white sports car and attempt to avoid the potholes. If you hit one or go off the road, the excitement is temporarily over.

Chapter 29

```
10 PRINT"[CLEAR,RVS,RED,SPACE8]POTHOLES - BRENT KAPILIK[SPACE8]"
15 POKE 53281,15:POKE 53280,15:POKE 53264,0:V=53248:POKE V+39,1
20 J=RND(-TI):S1=832:IF PEEK(65534)=23 THEN S1=3584:POKE 4861,1
25 FOR J=S1 TO J+64:POKE J,0:NEXT
30 POKE V+21,1:POKE 2040,S1/64:FOR J=0 TO 41:READ K:POKE S1+J,K:NEXT
35 PRINT"[CLEAR]":TI$="000000":IF PEEK(V+31)=1 THEN 35
40 FOR J=1 TO 6:PRINT SPC(40):NEXT
45 X=155:Y=59:POKE V,X:POKE V+1,Y
50 POKE V+21,1:FOR Z=1 TO 5
55 PRINT"[GREEN] [CMDR +11,YELLOW,CMDR C,SPACE14,CMDR V,GREEN,
    CMDR +11]"
60 IF PEEK(56320)=123 THEN X=X-15
65 IF PEEK(56320)=119 THEN X=(X+15) AND 255
70 POKE V,X:NEXT:IF PEEK(V+31)<>0 THEN POKE V+21,244:GOTO 80
75 P=INT(RND(0)*13)+13:PRINT"[UP]"TAB(P)"[BLACK,SHFT Q]":GOTO 50
80 PRINT"[DOWN,RVS,RED]POTHOLES - KAPILIK - JOYSTICK IN PORT 2"
85 PRINT"[DOWN,BLACK] YOUR SCORE= ";TI$:IF TI$>HS$ THEN HS$=TI$
90 PRINT "[SPACE3]HI SCORE= ";HS$:FOR J=1 TO 3000:NEXT:GOTO 35
91 DATA 000,000,000,001,254,000,003,003
92 DATA 000,003,003,000,003,255,000,001
93 DATA 254,000,001,002,000,001,002,000
94 DATA 001,002,000,001,254,000,003,255
95 DATA 000,003,255,000,003,051,000,001
96 DATA 254,000
```

Fig. 29-3.

Good Luck, and Happy Motoring!

Brett Kapilik
Winnipeg, Manitoba
Canada



Reach!

Figure 29-4 shows an unusual fast-action game with great play appeal and considerable addictive potential. It lacks color, sound, sprites and graphics, but it sure doesn't lack excitement!

In play, single moving digits appear along the bottom of the screen, waiting for you to launch them by pressing the Spacebar. When you press it, the moving digit jumps upward to the top of the screen, where it sticks to the first thing it touches. The object of the game is to amass a high score by positioning the digits along the top bar.

Your score on any "launch" is determined by the launched digit and by the product of the numbers it's touching horizontally and vertically once it lands. Any digit that isn't launched before it moves off the screen costs you 20 points.

```

100 T$="REACH! - VELAZQUEZ"
110 HX=10:KB=198:IF PEEK(65534)=23 THEN HX=12:KB=208
120 DIM M(41,18),A(40):H=(256*PEEK(251)+PEEK(252))*(PEEK(HX))
130 FOR N=1 TO 40:L$=L$+"[SHFT *]":NEXT
140 PRINT"[CLEAR,DOWN]";L$;"[DOWN20]"L$;
150 PRINT T$;"-[SPACE2,RVS]SPACEBAR[RVOFF] LAUNCHES"
160 POKE KB,0:D=48+INT(RND(0)*9+1):P=1:N=1864:IF RND(0)<.5 THEN P=-1
:N=1903
170 N=N+P:POKE N,D:POKE N-P,32:GET A$
180 IF A$=" " THEN C=ABS(N-1864):N=N-40:GOTO 210
190 IF P=1 AND N<1903 OR P=-1 AND N>1864 THEN 170
200 S=S-20:POKE N,32:Z=0:GOTO 260
210 A(C)=A(C)+1:E=1064+A(C)*40+C
220 FOR L=N TO E STEP-40:POKE L,D:POKE L+40,32:NEXT
230 X=C+1:Y=A(C):Z=D-48:M(X,Y)=Z
240 UP=M(X,Y-1):DN=M(X,Y+1):LF=M(X-1,Y):RG=M(X+1,Y)
250 IF Z=UP OR Z=DN OR Z=RG OR Z=LF THEN 280
260 S=S+Z*(DN+UP+LF+RG)+Z:IF S>H THEN H=S
270 PRINT"[HOME,SPACE4,LEFT4]";S,,"HI";H:GOTO 160
280 PRINT TAB(10)"[RVS]PLAY AGAIN (Y/N)"
290 GET A$:IF A$="" THEN 290
300 IF A$<>"Y" THEN PRINT"[CLEAR]GOODBYE![DOWN2]":PRINT"RUN[UP3]"
:END
310 HB=INT(H/256):LB=H-HB*256:POKE 251,HB:POKE 252,LB:POKE HX,1:RUN

```

Fig. 29-4.

The game ends when any digit lands next to or on top of an identical one.

Be careful when typing line 160 on a 64; if you add spaces or other extra characters, the line could exceed the 80-character limit.

*Carlos J. Velazquez
Hobe Sound, Florida*



Flower Power

No flowers to give to your sweetheart? Run Fig. 29-5 instead. Its products aren't as fragrant as the real thing, but they do just as well at conveying your sentiments.

*Bob & David Snader
Baltimore, Maryland*



Mighty Mite

This packs more punch in less space than any other program I can think of. Although it takes only three keystrokes to enter, it has a powerful effect that is totally different in 64 and 128 modes:

1 C [SHFT O]

When entering the program, you needn't put a space after the line number. And the [SHFT O] refers to the letter O, not to the numeral zero.

When you list the program on a 64 or Plus/4, you'll notice that the keyword has expanded to CONT. When you run it, you'll see that it locks up your entire system until the operator presses the Stop key. What other three-keystroke program can have such a powerful effect on computers, and on humans?

In 128 mode, the keyword expands to CONCAT. When you run the program, it makes your computer print an 18-character error message! Where else can you find a program that makes your computer print six times as many characters as you had to type to enter it?

But the power doesn't end here. If you do a FRE(0) before and after you enter the program, you'll find that this three-keystroke miracle occupies seven bytes of memory!

If you NEW the program and re-enter it with line number 12345, you'll see that the longer seven-keystroke program also occupies seven bytes. Will wonders never cease?

```

10 PRINT"[CLEAR,WHITE,SPACE4]FLOWER POWER - BOB & DAVE SNADER"
20 DIM A(64):V=53248:S=192:X=65534:IF PEEK(X)=23 THEN S=56:X=4861
30 PRINT"[DOWN2,GRAY3]GERMINATING...[ORANGE]":FOR L=64*S TO L+511
   :POKE L,0:NEXT
40 FOR T=0 TO 7:POKE 2040+T,S+T:NEXT
50 POKE X,1:FOR T=0 TO 29:READ P,B:POKE V+P,B:NEXT
60 PRINT"[DOWN16]":FOR Q=0 TO 63:READ N:A(Q)=N:NEXT
70 FOR T=0 TO 7:FOR Q=63 TO 0 STEP-1:L=L-1:POKE L,A(Q):FOR D=0 TO 40
80 NEXT:NEXT:NEXT
81 DATA 000,220,001,140,002,095,003,155
82 DATA 004,170,005,145,006,120,007,140
83 DATA 008,195,009,150,010,245,011,155
84 DATA 012,070,013,145,014,145,015,150
85 DATA 021,255,028,255,032,015,033,015
86 DATA 037,005,038,000,039,007,040,006
87 DATA 041,002,042,007,043,004,044,001
88 DATA 045,002,046,001,002,128,160,002
89 DATA 162,160,002,162,160,000,162,128
90 DATA 010,046,040,042,191,170,042,191
91 DATA 170,010,046,040,000,166,128,002
92 DATA 166,160,018,166,160,018,132,161
93 DATA 020,004,001,021,004,005,005,068
94 DATA 021,005,068,084,001,068,084,000
95 DATA 084,080,000,005,064,000,004,000
96 DATA 000,004,000,000

```

Fig. 29-5.


```

10 PRINT"[CLEAR,DOWN,SPACE9]LAMPMAN - ROBIN BOWNESS"
15 S=13:V=53248:X=65534:IF PEEK(X)=23 THEN S=56:X=4861
20 POKE 53280,0:POKE 53281,0:FOR J=64*S TO J+63:POKE J,255:NEXT
25 FOR J=J TO J+127:READ T:POKE J,T:NEXT
30 FOR J=0 TO 2:POKE 2040+J,S+J:NEXT:POKE X,1
35 POKE V+39,0:POKE V+40,5:POKE V+41,7
40 POKE V+21,7:POKE V+28,2:POKE V+37,7
45 POKE V+38,10:POKE V+3,120:POKE V+4,172
50 POKE V,150:POKE V+1,120:POKE V+2,150
55 POKE V+29,4:POKE V+5,133:FOR J=1 TO 20:PRINT:NEXT:FOR D=1 TO 50
: NEXT
60 FOR J=64*S+63 TO 64*S STEP-1:POKE J,0:FOR D=1 TO 50:NEXT:NEXT
61 DATA 001,085,000,005,253,064,007,051
62 DATA 064,023,255,080,031,207,208,015
63 DATA 255,195,043,003,015,043,003,138
64 DATA 170,206,170,042,170,168,010,170
65 DATA 160,042,138,160,010,170,160,002
66 DATA 162,160,002,170,160,000,168,160
67 DATA 000,170,160,000,042,168,000,010
68 DATA 168,000,042,168,000,000,010,223
69 DATA 000,000,000,000,000,000,000,000
70 DATA 000,000,000,000,000,000,000,000
71 DATA 000,000,000,000,000,192,000,000
72 DATA 224,000,012,048,060,030,028,126
73 DATA 051,015,255,195,007,255,195,007
74 DATA 255,195,001,255,102,000,254,060
75 DATA 000,254,024,000,124,000,000,056
76 DATA 000,001,255,000,007,255,192,007

```

Fig. 29-6.

△☆ Lampman

The fellow in Fig. 29-6's program has lots of power at his disposal. Be careful what you wish when he's about.

Robin Bowness
Pompano Beach, Florida

△☆ Jaws LXIV

Here's a challenge for all you game sharks. Run the program in Fig. 29-7, using the unshifted "<" and ">" keys to move Jaws left and right. Your object is to make him intercept the bubbles rising from the skindivers at the bottom of the sea (offscreen).

Each time the Terror of the Deep misses a bubble, he moves downward on the screen, making it harder for him to catch the next one. The game ends when he drops out of sight and into speargun range.

Chapter 29

```
10 PRINT"[CLEAR,WHITE]JAWS LXIV[GRAY3] - USE <&> - [YELLOW]
   BOB & DAVE SNADER"
15 L=1104:S=54296:TI$="000000":V=53248:X=150:Y=70:KY=197:SL=832
20 IF PEEK(65534)=23 THEN KY=213:SL=3584:POKE 4861,1:POKE 53264,0
25 POKE 53280,6:POKE 53281,6:POKE V+21,1:POKE V+39,1
30 FOR J=SL TO SL+128:POKE J,0:NEXT
35 FOR J=0 TO 33:READ Q,R:POKE SL+1+J,Q:POKE SL+65+J,R:NEXT
40 POKE L,32:POKE S,15:POKE S,0:L=1994+RND(8)*20:IF Y>250 THEN 80
45 K=PEEK(KY):M=M+(K=47):M=M-(K=44):X=X+2*M:POKE 2040,1+SL/64+(M<0)
50 IF X>255 THEN X=255:M=0
55 IF X<80 THEN X=80:M=0
60 POKE V,X:POKE V+1,Y:POKE L,32:POKE L-40,87:L=L-40
65 IF L<1100 THEN Y=Y+20:GOTO 40
70 IF PEEK(V+31) AND 1=1 THEN GOTO 40
75 PRINT"[HOME,DOWN,RIGHT2,L. GREEN]";:PRINT MID$(TI$,3,2);"
   :";RIGHT$(TI$,2);"[L. BLUE]":GOTO 45
80 PRINT"[WHITE]GAME OVER[L. BLUE]":FOR J=1 TO 10:GET A$:NEXT
81 DATA 002,064,000,000,000,000,012,048
82 DATA 000,000,000,000,060,060,000,000
83 DATA 003,128,252,063,001,192,031,096
84 DATA 254,127,006,248,119,057,255,255
85 DATA 156,238,255,031,255,255,248,255
86 DATA 127,031,255,255,248,254,031,056
87 DATA 254,127,028,248,000,096,248,031
88 DATA 006,000,000,128,048,012,001,000
89 DATA 000,000,008,016
```

Fig. 29-7.

An on-screen clock lets you test your survival skills against those of your friends.

Just when you thought it was safe to go back into the water, there's Jaws LXIV!

*Bob & David Snader
Baltimore, Maryland*

○△☆ Jumbles

The program in Fig. 29-8 creates puzzles similar to the popular Jumbles you see in the newspapers. Choose a topic and the number of Jumbles you want to produce, then stand back and see them come up on your printer.

The unscrambled words will be printed at the bottom of your sheet. If you don't want them, just delete the W\$(J) in line 270.

*Robert L. Sander
Pittsburgh, Pennsylvania*

```

100 PRINT"[CLEAR,RVS,SPACE10]JUMBLES - BILL SANDER[SPACE9]"
110 INPUT"[DOWN,SPACE6]TOPIC";T$
120 INPUT"[DOWN] # OF WORDS";N:IF N>25 THEN PRINT"[DOWN]
    25 WORDS MAX!":GOTO 120
130 DIM B(30),J$(N),L$(30),W$(N)
140 FOR J=1 TO 20:B$=B$+CHR$(164):NEXT
150 FOR W=1 TO N
160 PRINT"[DOWN,SPACE2]WORD #";W;:INPUT W$
170 L=LEN(W$):W$(W)=W$:FOR K=1 TO L:L$(K)=MID$(W$,K,1)
180 IF L$(K)=" " THEN L$(K)="-"
190 B(K)=K:NEXT
200 FOR K=1 TO L:D=INT(RND(0)*L+1):E=INT(RND(0)*L+1):T=B(D)
210 B(D)=B(E):B(E)=T:NEXT
220 FOR K=1 TO L:J$(W)=J$(W)+L$(B(K)):NEXT:NEXT
230 OPEN 4,4:CMD 4
240 PRINT SPC((80-LEN(T$))/2);T$:PRINT
250 PRINT:PRINT:FOR J=1 TO N:PRINT SPC(30-LEN(J$(J)));J$(J);"
    [SPACE2]";B$:PRINT:NEXT
260 FOR J=1 TO 53-3*N:PRINT:NEXT
270 FOR J=1 TO N:PRINT W$(J):NEXT
280 PRINT#4:CLOSE 4

```

Fig. 29-8.



Piece de Resistance

Berlitz defines this French term as "The principal dish at a meal. Also used figuratively." Language teachers seldom see the potential of current topics in electronics, so Berlitz never thought of this figurative use. Run the program in Fig. 29-9 and see what we mean.

Georg Simon Ohm
Erlangen, Germany

```

100 REM PIECE DE RESISTANCE - L.SANDER
110 A$="LUDWIG F. SANDER - 1986"
120 B$="[RVS,BROWN] [BLUE] [BROWN] [YELLOW] [BROWN] [ORANGE] [BROWN]
    ":C$="[SPACE10]"
130 D$="[SHFT M,DOWN,SHFT M,DOWN,SHFT M,DOWN,SHFT M]"
    :E$="[CMDR P10]"
140 F$=CHR$(206)+CHR$(205):G$=E$
150 H$="[SHFT N,UP,SHFT N,UP,SHFT N,UP,SHFT N,DOWN4]":I$="[BLACK]"
160 POKE 53281,13:PRINT"[CLEAR,DOWN3]";CHR$(142)
170 FOR J=1 TO 2:F$=F$+F$:NEXT
180 FOR J=1 TO 5:PRINT TAB(12);B$;C$:NEXT
190 PRINT"[UP3,SPACE2,RVS,GRAY3]";C$;SPC(10);"[BROWN]IRC[GRAY3
    RIGHT3]";C$
200 PRINT"[WHITE,SPACE2]";D$;E$;F$;G$;H$;TAB(58);"R 64";TAB(181);I$
210 PRINT TAB(11);"PIECE DE RESISTANCE"
220 PRINT TAB(59);"BY";TAB(89);A$;"[L. BLUE]"

```

Fig. 29-9.



Castle Siege

In Fig. 29-10's action-packed game, you besiege an enemy castle with the awesome power of a medieval catapult. Your boulders, if fairly launched and true, smash parts of the castle to dust. You can erase it completely with five perfect shots, but such devastating skill is only acquired through practice.

When you run the program, your catapult and the enemy fortress appear at the bottom of the screen, and you're asked to choose the elevation angle for your shot. Maximum distance is obtained at 45 degrees. Low elevations give flatter trajectories. Your catapult will not fire self-destructive vertical shots.

May your men be brave. May your weapon be strong. May your victory be total and swift.

*Bob & David Snader
Baltimore, Maryland*

```

10 PRINT"[WHITE,CLEAR,L. BLUE,RVS,SPACE4]
   CASTLE SIEGE - BOB & DAVE SNADER[SPACE4,L. GREEN] "
20 F=6:J=RND(-TI):Q=RND(TI)*16+1876:S=54296:V=53248
30 S1=2040:S2=832:IF PEEK(65534)=23 THEN S1=4861:S2=3584
40 FOR J=S2 TO J+64:POKE J,0:NEXT
50 DATA 000,091,039,127,040,241,041,255
60 DATA 076,127,077,241,078,255,080,194
70 DATA 117,194,118,160,119,251,120,160
80 FOR J=0 TO 11:READ L,M:POKE Q+L,M:NEXT
90 POKE 53281,0:POKE S1,13:POKE S2,128
100 POKE V+16,0:POKE V+29,1:POKE V+39,1
110 IF F=1 THEN PRINT"[DOWN4,L. RED] GAME OVER! YOU GOT[L. GREEN]";
   H;"[L. RED]POINTS.[DOWN2,L. BLUE]":END
120 POKE V+21,0:POKE 1985,105:POKE 1986,95:POKE 1945,70
130 X=38:Y=230:G=0:F=F-1
140 PRINT F;"[L. BLUE]SHOTS TO GO.":INPUT"[DOWN,GREEN]
   ELEVATION (1-88 DEG)";D:PRINT"[HOME,L. GREEN] "
150 IF D<1 OR D>88 THEN 140
160 R=[PI]/(180/D):A=TAN(R):B=3.3*COS(R)
170 POKE 1945,77:POKE V+21,1:POKE S,15:POKE S,0
180 X=X+B:Y=Y-(B*A)+G:G=G+.05
190 IF X>255 OR Y>250 THEN FOR J=0 TO 999:NEXT:GOTO 110
200 POKE V,X:POKE V+1,Y
210 IF PEEK(V+31)<>1 THEN 230
220 POKE 1024+INT((Y-50)/8)*40+INT((X-24)/8),32:H=H+1:POKE S,15
230 POKE S,0:GOTO 180

```

Fig. 29-10.

```

10 PRINT"[CLEAR,YELLOW,SPACE5]WHIZ BANG - BOB & DAVE SNADER"
15 DIM M(39),N(39):Q=1:R=255:S=54296:V=53248
20 S1=2040:S2=832:IF PEEK(65534)=23 THEN S1=4861:S2=3584
25 POKE S1,13:POKE V+16,0
30 FOR J=0 TO 63:POKE S2+J,0:NEXT
35 FOR J=0 TO 38:READ K,L:M(J)=K:N(J)=L:NEXT
40 FOR J=0 TO 4:READ K,L:POKE V+K,L:NEXT
45 POKE V,R-225*RND(0):Z=100*RND(0)+50
50 POKE V+39,Q:POKE S2+18,16
55 FOR Y=230 TO Z STEP-1:POKE V+1,Y:NEXT
60 POKE S,15:POKE S,0:POKE 53281,1:FOR J=1 TO 3:NEXT:POKE 53281,0
65 FOR J=0 TO 38:POKE S2+M(J),N(J):NEXT
70 FOR K=0 TO 300:NEXT:Q=Q+1:IF Q=8 THEN Q=1
75 FOR J=0 TO 38:POKE S2+M(J),0:POKE V+1,PEEK(V+1)+1
80 FOR K=0 TO 50:NEXT:NEXT:POKE S,5:POKE S,0:GOTO 45
81 DATA 019,040,016,017,022,016,013,130
82 DATA 025,131,010,041,028,040,015,007
83 DATA 017,192,007,108,006,030,008,240
84 DATA 004,068,009,097,011,012,012,128
85 DATA 014,002,001,131,000,003,002,128
86 DATA 018,056,020,056,021,064,023,004
87 DATA 024,129,026,002,027,006,029,192
88 DATA 031,068,034,068,030,008,032,032
89 DATA 037,198,033,016,035,016,040,130
90 DATA 036,016,038,016,043,130,033,000
91 DATA 021,001,032,000,023,001,029,001

```

Fig. 29-11.

△☆ Whiz Bang

Fire up your computer and see how Fig. 29-11 works! It has explosive sound, pyrotechnic graphics, and bang-up animation.

You'll have to type in the program, because we aren't revealing what it's all about (at least not directly).

*Bob & David Snader
Baltimore, Maryland*

○△☆ Psychological Insights

Nobody really understands the seductive lure of computing, but we who have succumbed are certain of its power. The following list appeared in a newspaper article on Gamblers Anonymous, a self-help organization for compulsive gamblers. They use it to identify prospective members.

For a spine-chilling thrill, type it into your word processor, then do a global search and replace, replacing “gambl” with “comput”.

- 1) Did you ever lose time from work due to gambling?
- 2) Has gambling ever made your home life unhappy?
- 3) Did gambling affect your reputation?
- 4) Have you ever felt remorse after gambling?
- 5) Did you ever gamble to get money with which to pay debts?
- 6) Did gambling cause a decrease in your ambition or efficiency?
- 7) Did you often gamble until your last dollar was gone?
- 8) Did you ever borrow to finance your gambling?
- 9) Have you ever sold anything to finance gambling?
- 10) Were you reluctant to use “gambling money” for normal expenditures?
- 11) Did gambling make you careless of the welfare of your family?
- 12) Did you ever gamble longer than you had planned?
- 13) Have you ever gambled to escape worry or trouble?
- 14) Did gambling cause you to have difficulty in sleeping?
- 15) Did you ever have an urge to celebrate any good fortune by a few hours of gambling?

Most compulsive gamblers will answer yes to at least five of these questions.

*John Procter
Las Vegas, Nevada*



Fuzzy Caterpillar

The program in Fig. 29-12 creates a most unusual “worm” on the screen. To control him, you must plug a joystick into Port 2. Note that all eight sprites are used to create one long, changing object. The program uses some interesting and unusual code to control the sprites and to do border checking.

Be sure to type the exact number of commas shown in line 30; they are essential to proper printing of the on-screen instructions.

*Scott Glazer
Reno, Nevada*

```

10 PRINT"[CLEAR,RVS,SPACE4]FUZZY CATERPILLAR - SCOTT GLAZER[SPACE4]"
15 V=53248:R=V+21:X=104:Y=104
20 S=832:IF PEEK(65534)=23 THEN S=3584:POKE 4861,1:POKE V+16,0
25 FOR T=0 TO 7:POKE T+2040,S/64:POKE V+39+T,7:NEXT
30 PRINT,,,"[DOWN6] IT WILL",,,," OBEY A",,,," JOYSTICK",,,,"
  " IN PORT 2"
35 FOR T=S TO S+63:READ A:POKE T,A:NEXT
40 J=PEEK(56320)
45 Y=Y+6*((J AND 1)=0) AND Y>55)-6*((J AND 2)=0) AND Y<225)
50 X=X+8*((J AND 4)=0)AND X>31)-8*((J AND 8)=0)AND X<248)
55 P=-P*(P<7)-(P<7):POKE R,255-2^P:POKE V+2*P,X:POKE V+2*P+1,Y
60 POKE R,255:GOTO 40
61 DATA 001,255,128,015,255,240,031,255
62 DATA 248,063,255,252,127,255,254,127
63 DATA 255,254,127,255,254,255,255,255
64 DATA 255,255,255,255,255,255,255,255
65 DATA 255,127,255,254,127,255,254,127
66 DATA 255,254,063,255,252,031,255,248
67 DATA 015,255,240,001,255,128,000,000
68 DATA 000,000,000,000,000,000,000,000

```

Fig. 29-12.



Defensive Driving

In the program in Fig. 29-13, you're defensively driving a space ship, and the game is really a winner. The action is so good that it upstaged my family's favorite commercial software for a time.

Your ship is the little blue block in the center of the screen, and you move it from side to side to avoid the enemy missiles. The longer you survive, the higher your score. Nobody survives very long.

Your controls are four keys, shown on the screen before every play.

When you move off the right side of the screen, you move one line closer to the bottom, which gives you less time to react. Moving off to the left moves you one line closer to the top. Touching the top or bottom of the screen will kill you.

Watch out for the teleport! If you land on a missile, you're dead. The rightward movements and the teleports occur at vastly different speeds in 64 and 128 modes. There must be a time warp in the ROMs.

Because of the superfast action involved, most games are very short. Fortunately, a single keypress gets you another one.

Steve Harney
El Paso, Texas

Chapter 29

```
100 T$="[HOME,RVS,SPACE4]DEFENSIVE DRIVING - STEVE HARNEY[SPACE4]"
110 KB=653:IF PEEK(65534)=23 THEN KB=211:T$=T$+" "
120 PRINT CHR$(158);CHR$(142);CHR$(8)
130 POKE 53280,0:POKE 53281,0:PRINT"[CLEAR]":GOTO 210
140 PRINT"[CLEAR]":FOR J=1 TO 50:PRINT:NEXT
150 B=32:C=54272:L=1524:T=TI:X=39
160 POKE L,120:POKE C+L,14:H=RND(1)*X
170 PRINT TAB(H);"[^]":L=L+PEEK(KB)*2-2
180 IF PEEK(L)=B THEN 160
190 POKE L,171:POKE C+L,14
200 S=TI-T:POKE 54296,15:POKE 54296,0
210 PRINT T$:PRINT"[UP,L. GREEN]YOUR SCORE=";S;
220 IF S>HS THEN HS=S
230 PRINT"[SPACE3]HIGH SCORE=";HS
240 PRINT"[DOWN]COMMODORE=RIGHT[SPACE4]CTRL=TELEPORT RIGHT"
250 PRINT"[SPACE3]NO KEY=LEFT[SPACE4]SHIFT=STRAIGHT[DOWN19]"
260 PRINT"[SPACE5]PRESS [RVS]Q[RVOFF] TO QUIT, [RVS]SPACE[RVOFF]
    TO PLAY[YELLOW]";
270 FOR J=1 TO 10:GET A$:NEXT
280 GET A$:IF A$=" " THEN 140
290 IF A$="Q" THEN END
300 GOTO 280
```

Fig. 29-13.



Chopper Pilot

Figure 29-14 is a dynamite game packed into just 22 lines of BASIC. It produces a full-size animated helicopter, complete with sound, which can be controlled by a joystick plugged into Port 2. Once you move the chopper in any direction, it keeps moving in that direction until you apply the appropriate amount of "reverse thrust." If you run into the border, you are smashed to smithereens.

Did you ever see so much action packed into such a little program?

*Bob & David Snader
Baltimore, Maryland*



64 Mirror

The program in Fig. 29-15 will definitely get the attention of the next person to use your 64. Just before you turn your computer over to the next person in line, load and run 64 Mirror.

*Frank Colaricci
Winter Park, Florida*

Amusements and Delights

```

10 PRINT"[CLEAR,L. RED,SPACE3]CHOPPER PILOT - BOB & DAVE SNADER"
15 V=53248:X=50:Y=237:S=13:IF PEEK(65534)=23 THEN S=56:POKE 4861,1
20 POKE V+16,0:POKE V+21,0:POKE V+29,1:POKE V+33,1:POKE V+39,6
25 FOR J=64*S TO J+8:READ K:POKE J,K:NEXT
30 FOR J=J TO J+29:READ K:POKE J,K:POKE J+64,K:NEXT
35 FOR J=J+25 TO J+8:READ K:POKE J,K:NEXT
40 J=PEEK(56320):L=L+(J=126)*.2:L=L+(J=125)*-.2
45 M=M+(J=123)*.2:M=M+(J=119)*-.2
50 X=X+M:Y=Y+L:B=(S+1)+(B=(S+1)):POKE 2040,B:POKE 54296,15
   :POKE 54296,0
55 IF F=0 AND X>255 THEN POKE V+16,3:X=0:F=1
60 IF F=0 AND X<24 OR Y>238 OR Y<50 THEN 80
65 IF F=1 AND X<0 THEN POKE V+16,0:X=255:F=0
70 IF F=1 AND X>40 OR Y>238 OR Y<50 THEN 80
75 POKE V+21,1:POKE V,X:POKE V+1,Y:GOTO 40
80 POKE 2040,1:FOR T=0 TO 200:Y=Y+1:IF Y>245 THEN Y=245
85 POKE V+1,Y:NEXT:RUN
90 DATA 000,000,000,000,007,000,000,002
91 DATA 000,224,006,000,144,031,128,072
92 DATA 114,064,063,242,032,007,254,032
93 DATA 000,255,192,000,031,128,000,010
94 DATA 000,000,018,064,000,063,192,007
95 DATA 224,000,000,031,192,000,002,063

```

Fig. 29-14.

```

10 PRINT"[CLEAR,RVS,SPACE6]64 MIRROR - FRANK COLARICCI[SPACE7]"
15 PRINT"[DOWN]THIS DOES TRICKERY WITH THE 64'S SCREEN."
20 PRINT CHR$(142);CHR$(8)
25 A=828:REM RELOCATABLE
30 FOR I=A TO A+87:READ B:POKE I,B
35 C=C+B:NEXT
40 IF C<>11844 THEN PRINT"ERROR":END
45 SYS A
50 DATA 173,014,220,041,254,141,014,220
51 DATA 165,001,041,251,133,001,169,000
52 DATA 133,251,133,253,169,208,133,252
53 DATA 169,048,133,254,162,008,134,250
54 DATA 160,000,177,251,162,008,106,038
55 DATA 002,202,208,250,165,002,145,253
56 DATA 200,208,239,230,252,230,254,198
57 DATA 250,208,231,165,001,009,004,133
58 DATA 001,173,014,220,009,001,141,014
59 DATA 220,173,024,208,041,240,009,012
60 DATA 141,024,208,169,048,133,056,096

```

Fig. 29-15.



64 Paratrooper

Figure 29-16's program uses two sprites that appear on the 64's screen as a paratrooper and his parachute. Not only is it fun to watch, but it illustrates the methods of defining, displaying and moving sprites on the 64.

If you have a 128, you can use the program in 64 mode or convert it to work in 128 mode as described in "64 Sprites on the 128," found in Chapter 28.

This brief analysis will help you understand just how the program works:

```

10 PRINT"[CLEAR,RVS] 64 PARATROOPER - RUSSELL GLENN HEIFNER "
61 DATA 000,129,000,001,129,128,003,060
62 DATA 192,006,086,096,006,187,096,006
63 DATA 070,096,006,060,096,003,153,192
64 DATA 001,195,128,000,219,000,000,219
65 DATA 000,000,090,000,000,024,000,000
66 DATA 195,000,001,255,128,001,195,128
67 DATA 003,129,192,003,129,192,001,129
68 DATA 128,001,195,128,007,195,224,000
69 DATA 000,000,000,000,000,000,000,000
70 DATA 000,000,000,000,000,000,000,000
71 DATA 000,000,000,024,000,000,126,000
72 DATA 000,255,000,001,255,128,003,255
73 DATA 192,015,255,240,031,129,248,060
74 DATA 000,060,056,000,028,048,000,012
75 DATA 008,000,016,004,000,032,002,000
76 DATA 064,001,000,128,000,129,000,255
100 POKE 53280,6:POKE 53281,6
110 PRINT"[HOME,DOWN22]";TAB(10);"[GRAY3,CMDR P20]"
120 PRINT TAB(10);"[RVS,CMDR A,SHFT *,CMDR S] PARATROOPER! [CMDR A,
SHFT *,CMDR S]"
130 PRINT TAB(10)"[RVS,CMDR Z,SHFT *,CMDR E,SHFT *14,CMDR E,SHFT *,
CMDR X,RVOFF,HOME]"
140 POKE 53248,160:POKE 53250,160:POKE 53277,3:POKE 53287,7
:POKE 53288,1
150 SL=15680:FOR J=SL TO SL+126:READ K:POKE J,K:NEXT
: REM SPRITES TO MEMORY
160 POKE 53269,3
170 POKE 2040,245:POKE 2041,246: REM SPRITE POINTERS
180 FOR Y=0 TO 191:POKE 53251,Y:POKE 53249,Y+20:NEXT
190 FOR J=1 TO 600:NEXT
200 POKE 2040,11:POKE 2041,11: REM SPRITE POINTERS
210 FOR Y=191 TO 250:POKE 53251,Y:POKE 53249,Y:NEXT
220 GOTO 170

```

Fig. 29-16.

Line 10 and lines 100-130 establish the text display on the screen. Notice how the landing platform at the bottom is composed of graphics available from the keyboard.

Line 140 establishes the initial position, color and size of the two sprites. Line 150 POKEs the sprites into memory, using data from lines 61-76. Line 160 "enables" the sprites, allowing them to be seen when they are positioned on the visible part of the screen.

Lines 170-220 control the paratrooper's motion from top to bottom of the screen.

Russell Glenn Heifner
Denver, Colorado



64 Sea Battle

See Fig. 29-17 for this one. You, the intrepid WWI flying ace, embark on a mission against an enemy ship. Your biplane gets ten flights, armed with one bomb each, to inflict sufficient damage to destroy the ship.

As you pass over the target, press any key to release your bomb. It will follow a normal trajectory affected by your altitude and speed, which are selected at random for each flight.

```

10 POKE 646,15:PRINT"[CLEAR] SCORE: 0[SPACE2,RVS]
    64 SEA BATTLE - THE SNADERS [RVOFF]"
15 POKE 915,16:Q=1774:S=54296:V=53248
20 FOR P=0 TO 9:READ L,M:A(P)=L:B(P)=M:NEXT
25 FOR T=0 TO 26:READ L:POKE 832+T,L:NEXT
30 POKE V+39,0:POKE V+40,1:POKE 2040,13:POKE 2041,14:POKE V+33,14
35 FOR T=0 TO 9:C=RND(TI)*(100)+65:B=C:K=0
40 Q=Q-2:FOR P=0 TO 9:POKE Q+A(P),B(P):NEXT:G=0:D=C/26:POKE V+21,3
    :POKE V+1,C
45 FOR X=0 TO 255 STEP D:POKE S,15:POKE S,0:POKE V,X:POKE V+2,X
50 GET K$:IF K$<>" " THEN K=1
55 G=G+(.2*K):B=B+G:POKE V+3,B:POKE V+32,7:IF B>212 THEN POKE V+21,1
    :K=0:G=0
60 IF PEEK(V+31) THEN H=H+1:PRINT"[HOME] SCORE:";H:POKE V+32,2
    :POKE V+21,1:K=0:G=0
65 NEXT:IF H>9 THEN PRINT"[CLEAR] SHIP DESTROYED":FOR T=0 TO 60
    :POKE V+33,T:NEXT:END
70 NEXT:PRINT" TRY AGAIN!"
75 FOR T=0 TO 24:FOR P=0 TO 9:POKE Q+A(P),B(P):NEXT:Q=Q-2:NEXT
80 DATA 000,103,001,032,002,032,038,095
81 DATA 039,247,040,160,041,098,042,098
82 DATA 043,032,044,032,224,127,192,240
83 DATA 008,128,248,005,004,063,255,228
84 DATA 007,255,252,000,127,228,000,003
85 DATA 132,000,000,128,000,001,192

```

Fig. 29-17.

Each bomb hit scores points, the total being displayed on the screen. Direct hits amidships score the highest, and it takes ten or more points to destroy the enemy ship. If you fail to destroy her, you watch helplessly as she makes her successful escape.

The game gets more difficult as the ship approaches the left of the screen, because you have less and less time to react to the relative motion of the plane and its target.

Important note: This program was prepared for a contest in which listings had to fit on a single screen; many keywords were typed with no spaces between them. But when the listing was prepared for print, spaces were inserted to make it easier to read. As a result, lines 40, 60 and 65 will exceed the 64's 80 character limit if you type them exactly as printed. When you encounter these lines, do not type the spaces between their keywords.

*Bob and David Snader
Baltimore, Maryland*



64 Timetrial

The green flag waves, and you're off! Three times around Fig. 29-18's track will prove what you're made of—three times around a viciously oil-slicked roadway!

Your car is a distinctly un-sporty purple square. You control it with a joystick in Port 2. Be warned that once the car starts moving on the slippery track, it takes an equal amount of reverse joystick to stop it. If you run off the track, the car will come to a stop, then respond very slowly until it's once again in the race . . . IF, I said IF (Heh, heh, heh)!

"64 Timetrial" comes complete with sound effects and a timing clock. In case you lose count during your frustration, a lap counter is displayed top left on the screen. When the counter reaches three, the race is over. Note your time, and press the fire button to race again.

A time of 40 seconds is very good. Get down to 35, and you're ready for Indianapolis!

Important note: "64 Timetrial" was written for a contest in which listings had to fit on a single screen—many keywords were typed with no spaces between them. But when the listing was prepared for print, spaces were inserted to make it easier to read. As a result, almost every one of the lines will exceed the 64's 80 character limit if you type them exactly as printed. When you encounter such a line, do not type the spaces between its keywords.

*Scott Glazer
Reno, Nevada*

```

10 PRINT"[CLEAR]";:V=53248:S=54272:POKE V+32,9:POKE V+33,0
  :REM 64 TIMETRIAL - S.GLAZER
20 FOR T=832 TO 958:POKE T,-255*(ABS(T-881.5)<41):NEXT:POKE 2040,13
  :POKE 2041,14
30 X=52:Y=136:POKE V+27,2:POKE V+21,3:POKE V+29,2:POKE V+39,4
  :POKE V+40,5:POKE V+2,42
40 POKE V+3,128:POKE V+21,3:Q=PEEK(V+31):PRINT"[CYAN]0[DOWN6]"
  :T$="[RVS,BROWN,SPACE16]"
50 T$=T$+"[SPACE6]":U$=LEFT$(T$,20):PRINT TAB(12)U$:FOR T=1 TO 10
  :PRINT TAB(10)T$
60 NEXT:PRINT TAB(12)U$:POKE S+24,15:POKE S+6,240:POKE S+4,49
  :TI$="000000"
70 POKE V+1,Y:POKE V+16,X/256:POKE V,X-PEEK(V+16)*256
  :Q=PEEK(V+30)=PEEK(V+31)
80 PRINT"[HOME,DOWN13,RVS]"TAB(18)INT(TI/6)/10:J=PEEK(56320)
  :F=9*(ABS(A)+ABS(D))
90 D=D+((J AND 1)=0)-((J AND 2)=0):A=A+((J AND 4)=0)-((J AND 8)=0)
  :X=X+A:Y=Y+D:POKE S,F
100 IF PEEK(V+31)AND 1 OR X<24 OR Y<47 OR Y>229 OR X>320 THEN POKE
  V+39,1:A=0:D=0:POKE V+39,4
110 IF C*PEEK(V+30)THEN C=0:L=L+1:POKE S-V,48+L
  :IF L=3 THEN WAIT 56320,16,16:RUN
120 C=C+(X>278)*(ABS(Y-160)<10):GOTO 70

```

Fig. 29-18.

**64 Break-Fast**

Figure 29-19 brings back the enjoyment of an arcade classic. To play the game, use a joystick in Port 2.

When you're typing the game, you must remove the spaces between the keywords in lines 110 and 220. Otherwise, they will exceed the 64's 80-character line length limit.

Though the game isn't fast enough for arcade hot shots, it requires concentration and fast reflexes. Your goal is to break through the wall of bricks at the top of the screen. You only have one ball, so there's little room for error.

If you think you're too good for a little game like this, try to beat my own high score of 8600 points.

Ghislain de Blois
 Timmins, Ontario
 Canada

**128 Screentron**

Figure 29-20 is the world's most unusual screen drawing program for the 128 and Plus/4.

Chapter 29

```
100 PRINT"[CLEAR, BLACK, SPACE5]64 BREAK-FAST[SPACE2]-[SPACE2]
G. DE BLOIS[DOWN2]"
110 FOR X=1 TO 40:A$=A$+"[RVS, SHFT O]":NEXT:POKE 53280,6
:POKE 53281,14:POKE 646,0
120 DX=1:DY=1:PX=38:PY=10:PP=18:PP$="[SHFT O, CMDR Y2, SHFT P]"
:PRINT"[BLACK]";A$;"[BLUE]";A$;
130 PRINT"[L. GREEN]";A$;"[YELLOW]";A$;"[WHITE]";A$;"[BLACK]"
140 PRINT"[HOME, DOWN22]";TAB(PP);PP$:ZZ=PX+40*PY:POKE 1024+ZZ,81
150 PRINT"[HOME, DOWN22]";TAB(PP);"[SPACE4]":POKE 54296+ZZ,0
160 X=(PEEK(56320) AND 15):IF X=11 THEN PP=PP-1:IF PP<0 THEN PP=0
170 IF X=7 THEN PP=PP+1:IF PP>36 THEN PP=36
180 PRINT"[HOME, DOWN22]";TAB(PP);PP$:POKE 1024+ZZ,32:PX=PX+DX
190 IF PX<=0 OR PX>=39 THEN DX=-DX
200 PY=PY+DY:IF PY<=0 THEN DY=-DY
210 ZZ=PEEK(1024+PX+40*PY):IF PY>23 THEN 250
220 IF ZZ=207 THEN SC=SC+100:POKE 54296,15:POKE 54296,0:DY=-DY
:GOTO 140
230 IF ZZ<>81 AND ZZ<>32 THEN DY=-DY:GOTO 140
240 IF SC<10000 THEN 140
250 PRINT"[HOME, DOWN9, RIGHT]GAME OVER!":PRINT"[RIGHT]YOUR SCORE =";
SC
```

Fig. 29-19.

```
100 REM 128 SCREENTRON - PETER BELOVSKY
110 REM USE JOYSTICK IN PORT 2
120 :
130 X=5:Y=5
140 GRAPHIC 1,1
150 COLOR 0,2:COLOR 4,16:COLOR 1,1
160 DRAW 1,X,Y TO Y,X
170 J=JOY(2):IF J=>128 THEN 130
180 IF J=1 OR J=2 OR J=8 THEN Y=Y-1
190 IF J=6 OR J=5 OR J=4 THEN Y=Y+1
200 IF J=8 OR J=7 OR J=6 THEN X=X-1
210 IF J=2 OR J=3 OR J=4 THEN X=X+1
220 IF Y<5 THEN Y=5
230 IF X<5 THEN X=5
240 GOTO 160
```

Fig. 29-20.

A joystick plugged into Port 2 controls the tiny dot in the upper left-hand corner of the screen. As you move the stick in various directions, a simulated three-dimensional "thing" will be drawn sloping downward to the right. Pressing the fire button erases everything and returns the dot to its home position.

Experiment with different directions on your joystick, and you'll see some unusual patterns develop.

Disable the program by simultaneously pressing Stop and Restore.

Peter Belovsky
New York, New York

☆ **128 Hidden Message**

Sometimes artists sign their work. Get into 128 mode and enter SYS 32800,123,45,6 to see an unusual example.

Kathleen Mead
Westerville, Ohio

☆ **128 Bird Art**

Figure 29-21 gives an interesting display on the 128's screen, plus some insight into the advanced use of sprites. It displays eight "birds" flying around on your screen.

Lines 130-150 draw ten-pixel-long lines at locations X and Y, at angles stored in A and B. These become the flapping wings. Line 190 puts the sprites in motion. Lines 220-230 flap the wings, and line 240 jumps back to line 200 where the birds change their direction.

Kevin Janzer
Address Unknown

```
100 REM 128 BIRD ART - KEVIN JANZER
110 A=0:B=360:L=10:X=10:Y=10:DIM A$(50)
120 FAST
130 DO:COLOR 0,15:COLOR 4,15:GRAPHIC 1,1
140 DRAW 1,X,Y TO L;B:DRAW 1,X,Y TO L;A:SSHAPE A$(N),0,0,20,20
:A=A+10:N=N+1
150 B=B-10:LOOP UNTIL B<0
160 L=1:DO:SPRITE L,1,1:L=L+1:LOOP UNTIL L>8
170 SLOW
180 GRAPHIC 0,1:L=1
190 DO:MOVSPR L,160,100:L=L+1:LOOP UNTIL L>8
200 DO
210 M=INT(RND(1)*360):FOR J=1 TO 8:MOVSPR J,M-(10*J)#5:NEXT
220 FOR B=0 TO N:SPRSV A$(B),1:SPRSV A$(B),2:SPRSV A$(B),3
:SPRSV A$(B),4
230 SPRSV A$(B),5:SPRSV A$(B),6:SPRSV A$(B),7:SPRSV A$(B),8:NEXT
240 LOOP
```

Fig. 29-21.

Chapter 29

```
100 PRINT"[CLEAR]128 EVASION - JOSE MIRANDA
110 FOR J=3601 TO 3646:POKE J,0:NEXT
120 FOR J=3602 TO 3627:READ K:POKE J,K:NEXT
130 SPRITE 1,1,2:CHAR 1,1,23:MOVSPR 1,170,100:COLLISION 2,180
140 J=JOY(2):K=INT(RND(8)*239)+1:PRINT TAB(K)"[SHFT Q]":S=S+5
150 IF J=7 THEN MOVSPR 1,270#3:SOUND 1,7500,1
160 IF J=3 THEN MOVSPR 1,090#3:SOUND 1,5000,1
170 GOTO 140
180 MOVSPR 1,0#0:PRINT"[HOME]GAME OVER -- YOUR SCORE IS:";S
190 FOR J=1 TO 3000:NEXT:PRINT"[CLEAR]":RUN 130
200 DATA 004,048,128,004,120,128,004,252
210 DATA 128,007,207,128,004,252,128,004
220 DATA 120,128,004,048,128,004,000,128
230 DATA 002,001
```

Fig. 29-22.

☆ 128 Evasion

Figure 29-22 is a nice game with lots of action in very few program lines. Use a joystick plugged into Port 2 to move your ship from side to side. You die when you collide with one of the meteors, but only for a few seconds.

If you're not yet a computer expert, you can use simple programs like this one to improve your programming skills as well as your piloting ability. A simple challenge would be to try adding statements to improve the colors on the screen. You have several to work with—border, background, titles, your ship and the invaders. Another challenge would be to add sound effects when a collision takes place, or to make the program keep track of the high score for the game.

*Jose M. Miranda
Jayuya, Puerto Rico*

○☆ 128 Commodore Logo

The program in Fig. 29-23 draws a colorful and familiar picture on your screen. Although it contains a lot of numbers, there's nothing mysterious about them. They're nothing more than screen locations and other parameters for the BASIC 7.0 statements used in the program.

If you have a monochrome monitor, it may be necessary to hold down Run/Stop and hit Restore before running the program. That sometimes makes the red show up more clearly.

*Jeff Johnson
Allensville, Pennsylvania*


```
10 REM 128 COMMODORE LOGO - JEFF JOHNSON
15 COLOR 0,16:COLOR 1,3:GRAPHIC 1,1
20 DRAW 1,294,136 TO 197,136 TO 197,104 TO 260,104 TO 295,136
25 PAINT 1,198,134
30 COLOR 1,7
35 CIRCLE 1,156,100,117,97,163,17
40 CIRCLE 1,156,100,60,49,145,35
45 DRAW 1,191,8 TO 191,61
50 DRAW 1,191,139 TO 191,192
55 PAINT 1,189,9
60 DRAW 1,294,64 TO 197,64 TO 197,96 TO 260,96 TO 295,64
65 PAINT 1,293,65
70 COLOR 4,16
Fig. 29-23.
```

☆ **128 Error Messages**

Enter the words **QUIT** or **OFF** in direct mode, and your 128 will surprise you with a message you've never seen before.

*John Kuhne
Newmarket, Ontario
Canada*

△☆ **Grand Finale**

Many readers were impressed with one of our fireworks programs by Bob and David Snader. I recently met the Snaders in person, and we discussed the possibilities of doing something more with the fireworks theme. They took the challenge and produced one of their usual gems. Here it is, in Fig. 29-24, courtesy of:

*Bob & David Snader
Baltimore, Maryland*

Related Tips in Other Chapters

Chapter 25's music and sound effects are also amusing and delightful.

Chapter 29

```
10 PRINT"[CLEAR,L. RED,DOWN2,SPACE4]GRAND FINALE - BOB & DAVE
  SNADER[SPACE4]"
15 S1=192:IF PEEK(65534)=23 THEN S1=56:POKE 4861,1
20 DIM L(39),P(39),A(16),B(16),C(16):S=54296:S1=S1*64:V=53248
25 POKE 53280,0:POKE V+33,0:POKE V+16,0:POKE V+21,255
30 FOR J=0 TO 38:READ Q,R:L(J)=Q:P(J)=R:NEXT
35 FOR J=S1 TO S1+510:POKE J,0:NEXT
40 FOR J=0 TO 15:A(J)=(RND(0)*8)+1:B(J)=RND(0)*255
45 C(J)=(RND(0)*150)+50:NEXT
50 FOR J=0 TO 7:POKE 2040+J,S1/64+J:POKE V+39+J,A(J):NEXT
55 POKE V+23,B(1):POKE V+29,B(1)
60 FOR J=0 TO 15:POKE V+J,C(J):NEXT
65 FOR K=S1 TO S1+510 STEP 64:POKE S,15:POKE S,0
70 FOR E=0 TO RND(0)*2:FOR J=0 TO 38:POKE K+L(J),P(J):NEXT:NEXT:NEXT
75 FOR J=0 TO 200:NEXT
80 FOR J=0 TO 38:FOR K=S1 TO S1+510 STEP 64:POKE K+L(J),0:NEXT:NEXT
85 GOTO 40
86 DATA 019,040,016,017,022,016,013,130
87 DATA 025,131,010,041,028,040,015,007
88 DATA 017,192,007,108,006,030,008,240
89 DATA 004,068,009,097,011,012,012,128
90 DATA 014,002,001,131,000,003,002,128
91 DATA 018,056,020,056,021,064,023,004
92 DATA 024,129,026,002,027,006,029,192
93 DATA 031,068,034,068,030,008,032,032
94 DATA 037,198,033,016,035,016,040,130
95 DATA 036,016,038,016,043,130
```

Fig. 29-24.

Useful Application Programs

*“’Tis not knowing much, but what is useful,
that makes a wise man.”*

Thomas Fuller
Gnomologia
1732

Application programs are those that serve non-computer purposes. They include things like checkbook balancers, games, word processors and spreadsheet programs.

Because they need to do useful work, it's hard to write a short one. This chapter shows how it's done.

○△☆ Lottery Assistant

Many states and provinces have lotteries in which the player chooses six numbers out of 40, or something in a similar range. The program in Fig. 30-1 can pick them for you at random. You could use it as a subroutine whenever you need to choose several numbers without repetition.

The program will pick any sized group of numbers from any group of equal or larger size, without giving duplicate picks. If the number of items picked equals the number of items picked from, the result is the same as shuffling the latter group.

If you win the lottery as a result of this program, please remember where you got it. Send your thanks to me, in care of Windcrest Books. Most winners also send a cashier's check for 15% of their gross.

```

1000 PRINT"[CLEAR,RVS,SPACE2]LOTTERY ASSISTANT - LOUIS F. SANDER
[SPACE3]"
1010 PRINT"[DOWN] PICKS SEVERAL NUMBERS, ALL DIFFERENT."
1020 PRINT"[DOWN] LINES 1040-1050 CHANGE THE PARAMETERS.[DOWN2]"
1030 REM INITIALIZATION - RUN ONCE ONLY
1040 A=40:REM # OF ITEMS TO PICK FROM
1050 B= 6:REM # OF ITEMS TO PICK
1060 DIM A(A),B(B)
1070 :
1080 REM PICK ITEMS - REPEAT AT WILL
1090 FOR J=1 TO A:A(J)=J:NEXT
1100 C=A:J=RND(-TI):IF B>A THEN STOP
1110 FOR J=1 TO B:K=INT(C*RND(1)+1)
1120 B(J)=A(K):A(K)=A(C):C=C-1:NEXT
1130 :
1140 REM DEMONSTRATES THE PICKS
1150 FOR J= 1 TO B:PRINT B(J);:NEXT
1160 PRINT:PRINT

```

Fig. 30-1.

○△☆ Game Map Maker

A good map of all the locations in an adventure game is very important to the serious adventurer. Figure 30-2's program, for use with any Commodore system and most compatible printers, will print 154 blank location squares on a single sheet of paper.

After typing in the program, save it to disk or tape. Run it and relax while the printer does its work. To use the maps, fill in the boxes with a short description of each known location. Draw arrows to show the

exits and entrances for each room. For up and down, I use a small letter "U" or "D" beside the arrow.

If a series of rooms can be reached only from one location, I find it easier to map these rooms separately from the main game map.

Bradley L. Dina
Euclid, Ohio

```

100 PRINT"[CLEAR,RVS,SPACE4]GAME MAP MAKER - BRADLEY L. DINA[SPACE4]
"
110 PRINT"[DOWN2] THIS PRINTS A SHEET FOR MAPPING GAMES."
120 PRINT"[DOWN2] READY THE PRINTER, THEN PRESS 'P'..."
130 GET A$:IF A$<>"P" THEN 130
140 B$=CHR$(255)+CHR$(26)
150 OPEN 4,4,10:CLOSE 4
160 OPEN 4,4:PRINT#4,CHR$(27);"P"
170 FOR Y=1 TO 14
180 PRINT#4,CHR$(8);
190 B=129:GOSUB 270
200 B=128:FOR J=1 TO 3:GOSUB 270:NEXT
210 B=192:GOSUB 270
220 FOR J=1 TO 3:PRINT#4:NEXT
230 NEXT
240 PRINT#4,CHR$(12):CLOSE 4
250 END
260 :
270 FOR X=1 TO 11:D=14:IF X=11 THEN D=10
280 GOSUB 290:NEXT:RETURN
290 PRINT#4,B$;CHR$(28);CHR$(B);B$;CHR$(D);CHR$(128);
300 RETURN

```

Fig. 30-2.

○△☆ Dice Machine

If you have played non-computer games, you've encountered those quaint little plastic things called "dice". Most such games use two six-sided dice, but some of the Dungeons and Dragons games use four-, eight-, ten- and twenty-sided dice, sometimes in quantities other than two.

It's often useful to computerize certain parts of these games, so more time can be spent playing and less spent bookkeeping. The program in Fig. 30-3 will give you the results of throwing any number of dice with any number of sides. You can use it as it is, or turn it into a subroutine to use in other programs.

Chapter 30

```
100 PRINT"[CLEAR,RVS,SPACE11]DICE MACHINE - TOM HYDE[SPACE6]"
110 X=RND(-TI)
120 INPUT"[DOWN,SPACE2]NUMBER OF DICE";D
130 INPUT" NUMBER OF SIDES";S
140 IF D<1 OR S<2 THEN PRINT"[DOWN] ???[DOWN]":GOTO 120
150 GOSUB 280
160 GET A$:IF A$="R" THEN 190
170 IF A$="Q" THEN 270
180 GOTO 160
190 FOR J=1 TO D
200 N=INT(RND(1)*S+1)
210 IF S>9 AND N<10 THEN PRINT" ";
220 PRINT N;:T=T+N:IF J<D THEN PRINT"+";
230 NEXT
240 PRINT"=";T
250 T=0:R=R+1:IF R=20 THEN R=0:GOSUB 280
260 GOTO 160
270 END
280 PRINT TAB(5);"[DOWN]PRESS [RVS]R[RVOFF] TO ROLL, [RVS]Q[RVOFF]
    TO QUIT[DOWN]":RETURN
```

Fig. 30-3.

If you use it elsewhere, be sure to keep the $X = \text{RND}(-TI)$ step. It's needed to ensure that you get a different sequence of numbers each time the program is run.

Tom E. Hyde
Camp Lejeune, North Carolina



Sprite Graph Maker

Designing sprite images is easier if you begin by sketching your ideas on graph paper. Figure 30-4 prints a numbered grid of 21×24 square blocks, just like those found in the 64 User's Manual.

Joseph R. Charnetski
Plains, Pennsylvania



64 Easy Word Processor

Here's a way to print out simple documents without loading your 64 word processor.

Create the document as you would a BASIC program, entering each line as a number followed by whatever text you want to print. Don't use PRINT or any other BASIC statements: what you type is what you will get. (Don't use any question marks, either, because the computer

will interpret them as PRINT statements.) You can change lines, delete and insert them, or anything else you could do with a real program.

When everything is as you want it, just execute this:

```
OPEN 4,4:POKE 22,35:CMD 4:LIST
```

This will list your bogus "program" to the printer, but without the line numbers. When the printer stops, enter:

```
PRINT#4:CLOSE 4:POKE 22,25
```

to return everything to normal. Be careful when typing PRINT#4: you cannot use the question mark as an abbreviation for PRINT, and you cannot include any spaces.

The program shown in Fig. 30-5 will automate much of the above process for you. If you run it before lines 10 and up have been added, it will prompt you to add them. Once lines 10 and up are part of the program, it will print them without their line numbers. If you save the program including these extra lines, it will be a self-printing word processor!

```
100 PRINT"[CLEAR,RVS] SPRITE GRAPH MAKER - JOSEPH CHARNETSKI "
```

```
110 PRINT"[DOWN2] PRESS 'P' TO PRINT A GRAPH, OR"
```

```
120 PRINT"[DOWN] PRESS 'Q' TO QUIT...[DOWN] "
```

```
130 GET A$
```

```
140 IF A$="Q" THEN END
```

```
150 IF A$<>"P" THEN 130
```

```
160 A$=CHR$(15):B$=CHR$(8):OPEN 4,4:CMD 4
```

```
170 FOR J=1 TO 3:PRINT"1";SPC(7);:NEXT:PRINT
```

```
180 FOR J=1 TO 3:PRINT"2631";SPC(4):NEXT:PRINT
```

```
190 FOR J=1 TO 3:PRINT"84268421";:NEXT:PRINT
```

```
200 FOR R=1 TO 21:FOR C=1 TO 24:PRINT CHR$(111);:NEXT
```

```
210 PRINT CHR$(165);MID$(STR$(R),2);B$:PRINT A$;:NEXT
```

```
220 FOR J=1 TO 24:PRINT CHR$(163);:NEXT
```

```
230 PRINT#4:CLOSE 4
```

Fig. 30-4.

```
0 PRINT"[CLEAR,RVS,SPACE2]64 EASY WORD PROCESSOR - TODD BRUSS
```

```
[SPACE3] "
```

```
1 IF FRE(0)<>-26975 THEN 7
```

```
3 PRINT"ENTER YOUR DOCUMENT AS A PROGRAM WITH"
```

```
4 PRINT"LINE NUMBERS STARTING AT 10.[DOWN2] "
```

```
5 FOR J=10 TO 100 STEP 10:PRINT J:NEXT:PRINT"[HOME,DOWN3] "
```

```
6 FOR J=0 TO 3:POKE 631+J,29:NEXT:POKE 198,4:CLR:END
```

```
7 PRINT"[DOWN2]OPEN 4,4:POKE 22,35:CMD 4:LIST 10-"
```

```
8 PRINT"POKE 22,25:PRINT#4:CLOSE4[HOME] "
```

```
9 FOR J=0 TO 2:POKE 631+J,13:NEXT:POKE 198,3:CLR:END
```

Fig. 30-5.

If the program doesn't work as described above, execute a PRINT FRE(0) with only lines 1-9 in memory. The computer will print a number on the screen, which you should then substitute for the - 26975 in line 1.

Todd Bruss
Hartford, Wisconsin



128 Easy Word Processor

Here's a way to print out simple letters or notes on your printer without loading your 128 word processor.

First, enter AUTO 10 to enable the 128's automatic line-numbering feature. Then type your letter just as you would a program, starting with line number 10. (The AUTO feature will automatically number the subsequent lines.)

When you've finished typing, execute these commands in direct mode:

```
OPEN 4,4:POKE 24,37:CMD 4:LIST
```

You will see your letter being printed without line numbers!

When the printing is finished, type this in direct mode:

```
PRINT#4:CLOSE 4:POKE 24,27
```

You can then save your writing as a program for further use and modification.

The secret to it all is the two POKEs to location 24. The first one disables the printing of line numbers, and the second enables them again.

When typing your letter, you should use all unshifted letters, and you must never use the question mark. (That's because the computer will interpret them as BASIC keywords such as PRINT when it's doing the listing.)

The program in Fig. 30-6 will automate the above procedure for you. If you run it before lines 10 and up have been added, it will establish the AUTO mode and start you typing on line 10. When you've finished typing, press Return when prompted for the next line number, then run the program again. This time, it will do the printing for you.

If the program doesn't work as described, execute a PRINT FRE(0) with only lines 1-5 in memory. You'll get a number on the screen, which you should then substitute for the 57907 in line 1.

You can save the entire program, including your typing and lines 1-5, and run it again to repeat the printout. You can add, delete, or change your lines of text, just as you would in a "real" word processing program.


```

0 REM 128 EASY WORD PROCESSOR - YANG
1 IF FRE(0)<57907 THEN 4
2 PRINT"[CLEAR,DOWN3]AUTO10":PRINT"[DOWN2]
  0 REM 128 EASY WORD PROCESSOR - YANG[HOME]"
3 POKE 842,13:POKE 843,13:POKE 208,2:END
4 OPEN 4,4:POKE 24,37:CMD 4:LIST 10-:PRINT#4:CLOSE 4:POKE 24,27:END
5 :

```

Fig. 30-6.

Wen-Ho Yang
Asuncion, Paraguay

○△☆ Label Printer

Those gummed labels with tractor feed holes are useful for more than mailing lists. Printed singly, they make ideal address labels for packages you send through the mail. Printed in large numbers with your own address, they make great return address labels. Multiple identical labels also come in handy if you send numerous letters to the same person. I keep a supply of such pre-addressed labels on hand: some with my own return address, some with the address of this magazine, some with my son's college address, and so on.

The short program in Fig. 30-7 is optimized for printing a single address repeatedly on the common $1\frac{5}{16}$ " \times $3\frac{1}{2}$ " labels, and is better for that purpose than any word processor. Because it uses upper and lowercase letters, before typing it in, put your computer into upper/lowercase mode by simultaneously pressing Shift and the Commodore key. To use the program, put the desired address in the six DATA lines numbered 301-306. If your address is shorter than six lines, as most of them are, put spaces or null strings into the unused DATA lines. When you run the program, it shows you the label it intends to print, giving you a chance to correct any errors. Then it prints a dummy label (the numbers 1-6) to help you align your paper. Finally, it prints any number of labels you ask for.

I keep several copies of the program on one disk, each with one of the addresses I use in my "stock" printed labels. Then when I need more labels, I load the appropriate program and make them. To make a "one of a kind" label, I load one of the programs, change the DATA statements, then run the program.

○△☆ Graph Paper Maker

If your printer can handle the graphics characters on the Commodore keyboard, Fig. 30-8 will create graph paper in almost any size you need.

Chapter 30

```
100 PRINT"[CLEAR,RVS,SPACE4,SHFT L]ABEL [SHFT P]RINTER - [SHFT L]
    OUIS [SHFT F]. [SHFT S]ANDER[SPACE5]"
110 PRINT CHR$(14);"[DOWN,SHFT T]HIS PRINTS ANY NUMBER OF THESE
    LABELS:"
120 FOR J=1 TO 35:PRINT"[CMDR O]";:NEXT:PRINT
130 FOR J=1 TO 6:READ A$(J):PRINT A$(J):NEXT
140 FOR J=1 TO 35:PRINT"[CMDR U]";:NEXT:PRINT
150 INPUT"[SHFT I]S THAT THE RIGHT LABEL (Y OR N)";A$
160 IF LEFT$(A$,1)="Y" THEN 190
170 PRINT"[DOWN,SHFT C]ORRECT THESE LINES, THEN [SHFT R,SHFT U,
    SHFT N] AGAIN."
180 LIST 301-:GOTO 280
190 PRINT"[DOWN,SHFT A]LIGN LABELS, TURN PRINTER ON, THEN"
200 PRINT"PRESS <RETURN> TO TEST ALIGNMENT..."
210 GET A$:IF A$<>CHR$(13) THEN 210
220 OPEN 4,4,7:FOR J=1 TO 6:PRINT#4,"[SHFT L]INE";J:NEXT
230 INPUT"[DOWN,SHFT I]S PRINT ALIGNMENT [SHFT O,SHFT K] (Y OR N)";
    A$
240 IF LEFT$(A$,1)="Y" THEN 260
250 PRINT"[DOWN,SHFT R]EALIGN LABELS, THEN [SHFT R,SHFT U,SHFT N]
    AGAIN.[DOWN]":GOTO 280
260 INPUT"[DOWN,SHFT H]OW MANY LABELS TO PRINT";N
270 FOR J=1 TO N:FOR K=1 TO 6:PRINT#4,A$(K):NEXT:NEXT
280 CLOSE 4:END
301 DATA " "
302 DATA "[SHFT L]OUIS [SHFT F]. [SHFT S]ANDER"
303 DATA "[SHFT P]OST [SHFT O]FFICE [SHFT B]OX 101011"
304 DATA "[SHFT P]ITTSBURGH, [SHFT P,SHFT A,SPACE2]15237"
305 DATA "[SHFT U].[SHFT S].[SHFT A]."
306 DATA " "
```

Fig. 30-7.

```
100 PRINT"[CLEAR,RVS]GRAPH PAPER MAKER - JOSEPH R. CHARNETSKI"
110 INPUT"[DOWN2]HOW MANY SQUARES ACROSS";X
120 IF X>39 THEN X=39
130 INPUT"[DOWN,SPACE2]HOW MANY SQUARES DOWN";Y
140 C1$="[CMDR G]":C2$=CHR$(8):C3$=CHR$(15):M=40-X
150 OPEN 4,4:PRINT#4:PRINT#4,SPC(M);
160 FOR ROW=1 TO Y : FOR COL=1 TO X
170 PRINT#4,"[SHFT O,CMDR Y]";:NEXT
180 PRINT#4,C1$;C2$:PRINT#4,C3$;SPC(M);
190 FOR J=1 TO X : PRINT#4,C1$;" ";:NEXT
200 PRINT#4,C1$;C2$:PRINT#4,C3$;SPC(M);:NEXT
210 FOR J=1 TO X*2 : PRINT#4,"[CMDR T]";:NEXT
220 PRINT#4:CLOSE 4
```

Fig. 30-8.

If your graph is less than 39 squares wide, it will be centered horizontally on the page.

When typing the program, be very careful NOT to overlook the many semicolons. Also, don't try to abbreviate the PRINT# statement.

Joseph R. Charnetski
Plains, Pennsylvania

○△☆ Wind Chill Calculator

Tell Fig. 30-9's program the temperature and the wind velocity, and it will tell you the "wind chill temperature."

Wind chill is a means of expressing the cooling effect of the combination of wind and low temperature, each of which causes a loss of heat. For example, if it's 10 degrees outside and the wind is blowing at 25 miles per hour, the cooling effect is the same as if it were -29 degrees with no wind. The -29 is referred to as the "wind chill temperature" for that combination of conditions. I tested this program against a chart in The World Almanac, and the two wind-chills never differed by more than one degree.

Billy D. Mullins
Houston, Texas

```

10 PRINT "[CLEAR,RVS]WIND CHILL CALCULATOR - BILLY D. MULLINS"
20 INPUT "[DOWN]TEMPERATURE IN DEGREES F";T
30 INPUT "[DOWN] WIND VELOCITY IN M.P.H.";V
40 T1=T:V1=(V*1609.35)/3600:TC=33-((T-32)*(5/9))
50 H=(10.45+(10*SQR(V1))-V1)*TC:X=H-506.784:IF X<0 THEN X1=T1
   :GOTO 70
60 X1=50-(X/12.3):X1=INT((X1*10)+5)/10
70 PRINT "[DOWN,SPACE2]WIND CHILL TEMPERATURE=";X1
80 PRINT:PRINT

```

Fig. 30-9.

○△☆ Fahrenheit to Celsius

Figure 30-10 is a handy science utility that will convert any Fahrenheit temperature to the more standard Celsius form. Be prepared for some long answers, because the conversion will be made to the ten millionths place on some numbers.

Although the program is user friendly, its operation is rather unsophisticated. Experimenters might want to add features like looping (for more temperatures) or reverse conversion from Celsius to Fahrenheit. It's not too hard to eliminate all those decimal places, either.

Chapter 30

```
10 PRINT"[CLEAR,RVS,SPACE2]FAHRENHEIT TO CELSIUS - BRETT MEYER
[SPACE3]"
20 PRINT
30 INPUT"TEMP IN DEGREES FAHRENHEIT";TF
40 TC=5/9*(TF-32)
50 PRINT
60 PRINT" THE CELSIUS TEMPERATURE IS";TC
```

Fig. 30-10.

It's always wise to do exhaustive testing with programs that do calculations. (How else will you know if the answers are right?) For temperature converters, I like to use 32, 212 and - 40 degrees Fahrenheit, which nicely convert to 0, 100 and - 40 degrees Celsius.

*Brett Meyer
Rota, Spain*



Batting Averages

Baseball and softball batting averages are always a topic of interest. Tell Fig. 30-11's program a player's at-bats and number-of-hits, and it will tell you his or her average. Can you figure out line 50?

As it stands, this is a very simple program, created to illustrate some elementary principles. You can test your programming ability by making it more sophisticated. You might want to add printer output, or to make the program respond properly if "YES" is entered instead of "Y".

*Shane Clark
Winchester, Indiana*

```
10 PRINT"[CLEAR,RVS,SPACE5]BATTING AVERAGES - SHANE CLARK[SPACE5]"
20 PRINT
30 INPUT"HOW MANY TIMES AT BAT";AB
40 INPUT"[SPACE8]HOW MANY HITS";H
50 AV=INT(1000*(H/AB)+.5)/1000
60 PRINT"[SPACE4]BATTING AVERAGE =";AV
70 PRINT
80 INPUT" FIGURE ANOTHER (Y/N)";A$
90 IF A$="Y" OR A$="[SHFT Y]" THEN 20
```

Fig. 30-11.



Alphabet Trainer

The simple program in Fig. 30-12 adds fun and animation to the task of learning the alphabet. As each letter appears on the screen, the

learner is required to search the keyboard for the letter matching the display. Pressing the correct key brings down the next letter in the alphabet.

Echoing the display from the keyboard should act to reinforce the learner's memory as well as physically involving him or her in the process.

If you simultaneously press the Shift and Commodore keys, the display will change to lowercase. The keyboard, of course, remains in uppercase. An advanced alphabet learner can develop knowledge of the two versions of each letter.

If it's hard to see the falling letter on your screen (as it is on the Plus/4), just substitute another color for the [YELLOW] at the end of line 110.

R. S. Stokan
Chicago Heights, Illinois

```

100 PRINT"[CLEAR,RVS,SPACE5]ALPHABET TRAINER - R. S. STOKAN[SPACE4]"
110 PRINT"[DOWN]PRESS THE KEY OF THE FALLING LETTER![YELLOW]"
120 FOR A=65 TO 90
130 GOSUB 200
140 GET A$:IF A$<>CHR$(A) THEN 140
150 NEXT A
160 PRINT"[HOME]":FOR J=1 TO 12:PRINT:NEXT
170 PRINT TAB(7);:FOR A=65 TO 90:PRINT CHR$(A);:NEXT
180 PRINT"[L. BLUE,DOWN7]"
190 END
200 PRINT"[HOME]":FOR J=1 TO 12
210 PRINT TAB(20);CHR$(A);"[UP]"
220 PRINT TAB(20);" ":NEXT
230 PRINT TAB(20);CHR$(A)
240 RETURN

```

Fig. 30-12.

```

100 PRINT"[CLEAR,RVS] DIVISION TUTOR - RICHARD L. EBERHARDY[SPACE2,
DOWN]"
110 J=RND(-TI):FOR J=1 TO 10
120 A%=10*RND(1)+1:B%=10*RND(1)+1:C%=A%*B%
130 PRINT"[DOWN]WHAT IS";C%;" / ";B%;:INPUT D
140 IF D=A% THEN PRINT"CORRECT!":E=E+1
150 IF D<>A% THEN PRINT"WRONG! THE ANSWER IS";A%
160 NEXT
170 PRINT"[DOWN]YOU GOT";E;" OUT OF 10 RIGHT, OR";E*10;"[LEFT]%"

```

Fig. 30-13.



Division Tutor

While there are many programs available to tutor children in simple arithmetic, many of them do not cover division. After trying to write such a program myself, I discovered why this is so—it's hard to find numbers whose quotient is an integer.

But the solution is simple. Because division is the inverse of multiplication, have the computer choose two random integers and multiply them together. Then for your division problem, divide that product by one of the two original numbers. The program shown in *Fig. 30-13* illustrates the principle.

Richard L. Eberhardy
Green Bay, Wisconsin

Disk Utility Programs

*“Nothing can have value
without being an object of utility.”*

Karl Marx
Capital
1867-1883

Utility programs are those whose work is related to the computer system itself.

The most valuable (and most numerous) utility programs are those that relate to the disk drive. This chapter has some of the best of them.



File Renamer

Although renaming files is easy to do from direct mode, the syntax can be confusing and difficult to remember. Figure 31-1's program automates renaming by prompting you for the old and new filenames, then doing the confusing stuff itself.

*Toby Dodd
Robinson, Illinois*

```

100 PRINT"[CLEAR,RVS,SPACE8]FILE RENAMER - TOBY DODD[SPACE8]"
110 INPUT"[DOWN]DO YOU WISH TO RENAME A FILE (Y/N)";A$
120 IF A$<>"Y" THEN PRINT "[DOWN]I'LL INTERFACE WITH YOU LATER,
    THEN![DOWN2]":END
130 INPUT"[DOWN]OLD FILENAME";OF$
140 INPUT"[DOWN]NEW FILENAME";NF$
150 OPEN 15,8,15
160 PRINT#15,"R0:" +NF$+"="+OF$
170 CLOSE 15
180 GOTO 110

```

Fig. 31-1.



Directory Viewer

The routine in Fig. 31-2 displays the disk directory on your screen without loading it into the computer's memory. If you use it in another program as a subroutine, you'll have instant access to the directory at any time while your main program is running. You can easily exit by pressing Q to quit.

Although "Directory Viewer" works perfectly on the 128 and Plus/4, owners of those machines have access to the DIRECTORY command, which performs the same function with no need for extra programming.

*Joseph R. Charnetski
Plains, Pennsylvania*

```

100 REM DIRECTORY VIEWER - J.CHARNETSKI
110 REM SHOWS DISK DIRECTORY ON SCREEN
120 OPEN 1,8,0,"$":GET#1,A$,A$
130 GET#1,A$,A$:IF A$="" THEN 190
140 GET#1,B$,C$
150 BL=ASC(B$+CHR$(0)):BH=ASC(C$+CHR$(0))
160 D$=MID$(STR$(BL+256*BH),2)+CHR$(32)
170 PRINT D$;:GET#1,D$:IF D$<>"" THEN 170
180 PRINT:GET E$:IF E$<>"Q" THEN 130
190 CLOSE 1:END

```

Fig. 31-2.

○△☆ **Blocks Free Reader**

Figure 31-3 determines the number of blocks free on the disk, putting that number into the variable F. You can use it as a subroutine in another BASIC program.

Geoff Larsen
Kamloops, British Columbia
Canada

```

100 REM BLOCKS FREE READER - LARSEN
110 REM SHOWS DISK BLOCKS FREE ON SCREEN
120 DEF FN A(X)=ASC(A$+CHR$(0))+256*(ASC(B$+CHR$(0)))
130 OPEN 1,8,0,"$0":GET#1,A$,B$
140 GET#1,A$,B$:IF FN A(X)=0 THEN 180
150 GET#1,C$,D$
160 GET#1,E$:IF ASC(E$+CHR$(0)) THEN 160
170 GOTO 140
180 A$=C$:B$=D$:F=FN A(X):CLOSE 1
190 PRINT F;"BLOCKS FREE ON DISK"

```

Fig. 31-3.

○△☆ **Write-Protect Detector**

Figure 31-4 will detect if there's a write-protect tab on the disk in drive 8. Although it's written as a program, you'd normally use it as a subroutine.

To do that, change the END in line 2040 to a RETURN.

Joseph R. Charnetski
Plains, Pennsylvania

```

2000 REM WRITE-PROTECT DETECTOR - JOSEPH R. CHARNETSKI
2010 OPEN 1,8,15:PRINT#1,"M-R";CHR$(30);CHR$(0):GET#1,A$:CLOSE 1
2020 IF ASC(A$+CHR$(1))=1 THEN PRINT "DISK IS WRITE-PROTECTED"
      :GOTO 2040
2030 PRINT"DISK IS NOT WRITE PROTECTED"
2040 END

```

Fig. 31-4.

○△☆ **Start Address Finder**

Have you ever forgotten the SYS address of a machine language program? If you have, Fig. 31-5 may save the day.

Put a disk with your machine language program into drive 8, then run "Start Address Finder" and enter the name of the machine language program. The program's starting address will be printed onto the screen.

```

10 REM START ADDRESS FINDER - K.CARROLL
20 INPUT "[SPACE2]FILENAME";F$
30 OPEN 2,8,2,F$:GET#2,A$,B$:CLOSE 2
40 PRINT"START ADDR=";
50 C$=CHR$(0):PRINT ASC(A$+C$)+256*ASC(B$+C$)

```

Fig. 31-5.

The starting address is the lowest memory location used by the machine language program; nine times out of ten, that's also the address you SYS to start the program.

“Start Address Finder” works with BASIC programs, too.

*Kennet Carroll
Lanett, Alabama*



End Address Finder

Figure 31-6 finds the end address of any program on your disk.

```

100 PRINT"[CLEAR,RVS,SPACE2]END ADDRESS FINDER - LOUIS F. SANDER
[SPACE2]"
110 PRINT"[DOWN]FINDS THE END ADDRESS OF A DISK FILE."
120 Y=782:IF PEEK(65534)=23 THEN Y=8
130 INPUT"[DOWN]FILENAME";F$:IF F$="" THEN 200
140 OPEN 2,8,2,F$
150 POKE Y,1:SYS 65493
160 CLOSE 2
170 OPEN 15,8,15:INPUT#15,A,B$:CLOSE 15:IF A>19 THEN PRINT"[DOWN,
RVS]";B$:GOTO 200
180 EA=PEEK(174)+256*PEEK(175)
190 PRINT"[DOWN]END ADDRESS IS";EA
200 END

```

Fig. 31-6.



Memory Saver

The program in Fig. 31-7 will save a machine language program, or any other range of memory, to a disk in drive 8. It's been tested on the 64, but should work on any Commodore machine. If you try it on the 128, you may have to add some BANK statements.

The routine can be used as a stand alone program or as a subroutine. If you use it in another program, you can replace the INPUT statements with others to specify the addresses and filename.

Used with a long BASIC program that pokes machine language into place, this routine can make a quick-loading version of the machine

language program. I've also used it to save text and hi-res screens for quick reloading from disk.

When you type it in, don't overlook the semicolons at the end of lines 180 and 190.

Robert Bixby
Kalamazoo, Michigan

```

100 PRINT"[CLEAR,RVS,SPACE6]MEMORY SAVER - ROBERT BIXBY[SPACE7]"
110 PRINT"[DOWN]THIS SAVES A MEMORY RANGE TO DISK."
120 INPUT"[DOWN]START ADDRESS";A
130 INPUT"[SPACE2]END ADDRESS";B
140 A1=INT(A/256):A2=A-256*A1
150 INPUT"[SPACE5]FILENAME";E$
160 OPEN 15,8,15,"S0:"+E$
170 OPEN 1,8,1,"0:"+E$+",P,W"
180 PRINT#1,CHR$(A2)CHR$(A1);
190 FOR J=A TO B:PRINT#1,CHR$(PEEK(J));
200 NEXT
210 CLOSE 1:VERIFY E$,8,1
220 INPUT#15,A$,B$:PRINT A$,B$:CLOSE 15
230 END

```

Fig. 31-7.



64 ML Saver

It can be quite time-consuming to load those BASIC programs that contain machine language programs in DATA statements. After a little initial setup, it's faster and easier to load the machine language directly.

128 owners can use BASIC 7.0's BSAVE and BLOAD statements to directly SAVE and LOAD machine language. When you read the System Guide's explanations, remember that "binary file" means the same as "machine language file."

As an example of how this can be done on the 64, consider the "Magazine Entry Program 64," found in the back of each month's *Commodore Magazine* and on the disk you can buy to accompany this book. When it is run, this BASIC program creates a machine language program located in memory locations 49152-49900. The machine language program does all the work, and the large, slow-to-execute BASIC loader is needed only to get the machine language into the computer. But once the machine language is in memory, it's a simple matter to save it directly. The resulting file will load and execute much faster than the original BASIC.

To save the machine language on the 64, we must convert its starting and ending addresses into low byte-high byte format. Then we POKE

those addresses into the memory locations that the SAVE command uses in determining what locations to save. Finally, we execute the SAVE command to save the machine language to disk.

The program in Fig. 31-8 can be used to automate the process. Save it before you use it, because it self-destructs when it is run. Be careful when typing line 200; if you insert extra spaces or other letters, its length may exceed 80 characters, and you will not be able to enter it.

To use the "64 ML Saver" program, get your machine language into memory, taking note of its starting and ending addresses. Then load and run "64 ML Saver," assigning a unique filename to the machine language you want to save.

When the program is finished, you'll be asked to execute three statements to save your machine language. The statements will be printed on the screen, with the cursor flashing on the first one. Just press Return three times, and the save will be accomplished.

Because this process corrupts some of BASIC's internal workings, you will have to reset your computer as soon as it's finished. If you don't have a Reset button, you'll have to turn the computer off to reset it.

From now on, you can reload the machine language by executing LOAD "filename",8,1. Once it has been loaded, you must execute a direct-mode NEW command to prepare the 64 for BASIC.

If the NEW disturbs you, you can load the ML from within a BASIC program by using the following line at the start of the BASIC:

```
10 IF A=0 THEN A=1 : LOAD "filename",8,1
```

```
100 PRINT"[CLEAR,RVS,SPACE6]64 ML SAVER - ROLAND TOBIASZ[SPACE6]"
110 INPUT"[DOWN]FIRST ADDRESS TO SAVE";FA
120 IF FA<0 OR FA>65535 THEN 110
130 H1=INT(FA/256):L1=FA-256*H1
140 INPUT"[DOWN] LAST ADDRESS TO SAVE";LA
150 LA=LA+1:Q$=CHR$(34)
160 IF LA<FA OR LA>65535 THEN 140
170 H2=INT(LA/256):L2=LA-256*H2
180 INPUT"[DOWN]FILENAME";F$:L=LEN(F$)
190 IF L<1 OR L>16 THEN 180
200 PRINT"[DOWN]TO SAVE YOUR ML, EXECUTE THESE COMMANDS,
    THEN RESET YOUR COMPUTER:[DOWN2]"
210 PRINT"POKE 43,";L1;" :POKE 44,";H1
220 PRINT"[DOWN2]POKE 45,";L2;" :POKE 46,";H2
230 PRINT"[DOWN2]SAVE";Q$;F$;Q$;" ,8,1"
240 PRINT"[UP11]"
```

Fig. 31-8.

The IF...THEN is required because of Commodore BASIC's unusual handling of program-mode LOADs.

Roland Tobiasz
Nowy Sacz, Poland



64 Memory Saver

Figure 31-9 can be used to save any area of 64 memory to disk. You may find it useful in saving the machine language generated by the other programs in this book.

To use "64 Memory Saver" put your machine language, sprite, or whatever into memory and note its starting and ending addresses. Choose a filename under which to save it, then run this program.

When you respond to the prompts for filename, start address and end address, the indicated area of memory will be saved from the start address to end address inclusive.

If the disk contains another file with the same filename, that file will be scratched before the new one is saved. With this feature, you can save your updates as you make them, without having to give them unique filenames.

Michael J. Gibbons
Orange City, Florida

```

100 PRINT"[CLEAR,RVS,SPACE5]64 MEMORY SAVER - MIKE GIBBONS[SPACE5]"
110 PRINT"[DOWN]SAVE W/REPLACE ANY SECTION OF MEMORY"
120 INPUT"[DOWN2,SPACE7]DRIVE#[SPACE2]8[LEFT3]";D
130 S$=" ":F$=" ":INPUT"[SPACE5]FILENAME";F$:IF F$="" THEN 130
140 INPUT"START ADDRESS";A1:INPUT"[SPACE2]END ADDRESS";A2
150 OPEN 15,8,15,"S0:"+F$:CLOSE 15
160 A$=F$:L=PEEK(53)+256*PEEK(54)-LEN(A$):POKE 782,L/256
170 POKE 781,L-PEEK(782)*256:POKE 780,LEN(A$):SYS 65469
180 POKE 780,1:POKE 781,D:POKE 782,1:SYS 65466
190 T=A1:POKE 254,T/256:POKE 253,T-PEEK(254)*256:POKE 780,253
200 T=A2+1:POKE 782,T/256:POKE 781,T-PEEK(782)*256:SYS 65496
210 OPEN 15,8,15:INPUT#15,A,B$,C$,D$:CLOSE 15
    :PRINT"[DOWN]";A;B$;S$;C$;S$;D$

```

Fig. 31-9.



SEQ File Reader

Figure 31-10 is an amazing one-liner that will read any sequential file and print its contents to the screen. Just replace the word FILENAME with the actual name of the file, then execute the program. If your file

has a long name, you may have to delete some of the spaces between words in the program line, to limit its length to the 80 character maximum for a 64.

*John Matosky
Chicopee, Massachusetts*

```
10 REM SEQ FILE READER - JOHN MATOSKY
20 REM  READS SEQ FILE 'FILENAME', AND
30 REM  PRINTS IT TO THE SCREEN.
40 OPEN 8,8,8,"FILENAME":FOR J=0 TO 1:GET#8,A$:J=ST:PRINT A$;:NEXT
   :CLOSE 8
```

Fig. 31-10.



128 SEQ File Reader

Figure 31-11 may be the world's shortest SEQ file reader. The work is done by pre-existing ROM routines in the 128. To use the program, substitute the name of your SEQ file for FILENAME.

*E. G. Bell
Pittsburgh, Pennsylvania*

```
1 REM 128 SEQ FILE READER - E.G.BELL
2 INPUT"FILENAME";F$:OPEN 0,8,0,F$+",S,R":SYS 41149
```

Fig. 31-11.



Disk Signer

Leave your mark on all of your disks as a friendly reminder to those who borrow them. Just run Fig. 31-12 on the desired disk before putting any other files on it. Your signature can't be removed from the disk by normal means.

*Thong Nguyen
Panama City, Florida*



64 Directory Basher

The nifty program in Fig. 31-13 will make your disk directory unlistable on the blue default screen of a 64, even with the "\$" command from DOS 5.1 or the Fast Load cartridge. As an added feature, it lets you restore a bashed directory to its normal state.

*Thong Nguyen
Panama City, Florida*

```

100 PRINT"[CLEAR,RVS,SPACE7]DISK SIGNER - THONG NGUYEN[SPACE7]"
110 PRINT"[DOWN]PRESS 'S' TO SIGN YOUR NAME ON A NEW"
120 PRINT"DISK, OR PRESS ANY OTHER KEY TO QUIT."
130 GET A$:IF A$="" THEN 130
140 IF A$<>"S" THEN END
150 PRINT"[DOWN]CHECK THE DISK, THEN PRESS 'D'."
160 GET A$:IF A$<>"D" THEN 160
170 FOR X=1 TO 16:F$=F$+CHR$(160):NEXT
180 O$="PROPERTY OF"
190 OPEN 15,8,15:OPEN 5,8,5,"#"
200 INPUT"[DOWN]YOUR NAME (1-12 CHARS)";N$
210 N$=LEFT$(N$,12):S=0:GOSUB 260
220 P=144:M$=CHR$(141)+CHR$(147):GOSUB 270
230 S=1:GOSUB 260:P=2:X$=O$:GOSUB 290:GOSUB 270
240 P=34:X$=N$:GOSUB 290:GOSUB 270
250 PRINT#15,"I0":END
260 PRINT#15,"U1";5;0;18;S:RETURN
270 PRINT#15,"B-P";5,P:PRINT#5,M$;
280 PRINT#15,"U2";5;0;18;S:RETURN
290 M$=CHR$(232)+CHR$(0)+CHR$(0)
300 M$=M$+CHR$(160)+CHR$(20)+CHR$(20)
310 M$=M$+CHR$(18)+X$+F$:M$=LEFT$(M$,19)
320 RETURN

```

Fig. 31-12.

```

100 PRINT"[CLEAR,RVS,SPACE3]64 DIRECTORY BASHER - THONG NGUYEN
[SPACE3]"
110 PRINT"[DOWN2]'BASHING' MAKES A DIRECTORY LIST IN"
120 PRINT"BLUE, SO YOU CANNOT SEE IT ON A 64'S"
130 PRINT"NORMAL BLUE SCREEN."
140 PRINT"[DOWN](B)ASH, (U)NBASH OR (Q)UIT?"
150 GET A$:IF A$="B" OR A$="U" THEN 180
160 IF A$="Q" THEN END
170 GOTO 150
180 OPEN 15,8,15:OPEN 5,8,5,"#"
190 PRINT#15,"U1 5 0 18 0":IF A$="B" THEN 210
200 FOR X=1 TO 6:M$=M$+CHR$(160):NEXT:GOTO 230
210 FOR X=1 TO 3:M$=M$+CHR$(20):NEXT
220 M$=M$+CHR$(141)+CHR$(31)+CHR$(147)
230 PRINT#15,"B-P";5;144:PRINT#5,M$;
240 PRINT#15,"U2 5 0 18 0":PRINT#15,"I"

```

Fig. 31-13.

○△☆ File Goofer

Run Fig. 31-14 to zero out all the file length indicators in your directory. When used on a 64 with the 64 "Directory Basher" program, it has a great effect as a security measure.

"File Goofer" also works on the 128.

Thong Nguyen
Panama City, Florida

```

100 PRINT"[CLEAR,RVS,SPACE7]FILE GOOFER - THONG NGUYEN[SPACE7]"
110 PRINT"[DOWN]PRESS '0' TO SET ALL FILE LENGTHS TO 0,"
120 PRINT"OR PRESS ANY OTHER KEY TO QUIT."
130 GET A$:IF A$="" THEN 130
140 IF A$<>"0" THEN END
150 PRINT"[DOWN]CHECK DISK, THEN PRESS 'D'"
160 GET A$:IF A$<>"D" THEN 160
170 OPEN 15,8,15:OPEN 5,8,5,"#":FOR C=1 TO 3
180 FOR S=C TO C+15 STEP 3:GOSUB 200:NEXT:NEXT
190 PRINT#15,"I0":END
200 PRINT#15,"U1:";5;0;18;S:GET#5,A$
210 IF A$="" THEN A$=A$+CHR$(0)
220 IF ASC(A$)=0 THEN GOSUB 230:GOTO 190
230 FOR B=30 TO 254 STEP 32
240 PRINT#15,"B-P";5;B:PRINT#5,CHR$(0);
250 NEXT:PRINT#15,"U2:";5;0;18;S:RETURN

```

Fig. 31-14.

△ 64 Disk Defaulter

Are you tired of typing ,8 every time you want to use your disk drive? Would you spend five minutes to get permanent relief? If so, type in the program from Fig. 31-15, which changes the default device number from 1 to 8.

```

10 PRINT"[CLEAR,RVS,SPACE4]64 DISK DEFAULTER - RICHARD PENN[SPACE4]"
20 PRINT"[DOWN]CHANGES THE DEFAULT DEVICE NUMBER TO 8."
30 PRINT"[DOWN]STOP/RESTORE RESTORES DEFAULT TO 1."
40 PRINT"[DOWN]SYS 679 CHANGES IT BACK TO 8."
50 FOR J=679 TO 716:READ K:POKE J,K:CS=CS+K:NEXT
60 IF CS<>4119 THEN PRINT"ERROR IN DATA!":STOP
70 SYS 679:NEW
71 DATA 169,002,141,049,003,141,051,003
72 DATA 169,186,141,048,003,169,197,141
73 DATA 050,003,096,169,008,133,186,169
74 DATA 000,133,010,076,165,244,169,008
75 DATA 133,186,076,237,245,010

```

Fig. 31-15.

Be sure to save it before you use it the first time, because using it erases it!

Pressing Stop/Restore will disable the utility, while SYS 679 will bring it back to life.

*Richard Penn
Montreal, Quebec
Canada*



64 Disk Error Reader

If you're tired of mysterious error lights on your disk drive, the program in Fig. 31-16 is just the thing you need.

When run, it puts a machine language program into the tape buffer, where it stays until you use the tape drive or reset the computer. When your error light blinks, call the machine language program by typing SYS 830. It responds by turning off the light and displaying the message from the disk error channel.

If you like, you can add a NEW to the end of the BASIC program; then it will remove itself from memory after it's run, leaving the machine language to do the job you need.

By the way, the machine language is completely relocatable—if you know about such things, you can put it somewhere else in memory.

*Faisal Akbar Ali
Dubai, United Arab Emirates*

```

10 PRINT"[CLEAR,RVS]64 DISK ERROR READER - FAISAL AKBAR ALI "
20 FOR J=830 TO 868:READ K:POKE J,K:CS=CS+K:NEXT
30 IF CS<> 5736 THEN PRINT"ERROR IN DATA STATEMENTS!":STOP
40 PRINT"[DOWN]AFTER THIS HAS BEEN RUN, SYS 830 STOPS"
50 PRINT"THE DISK LIGHT BLINKING AND DISPLAYS"
60 PRINT"THE DISK ERROR MESSAGE.[DOWN2]"
61 DATA 169,015,168,162,008,032,186,255
62 DATA 169,000,032,189,255,032,192,255
63 DATA 162,015,032,198,255,032,207,255
64 DATA 201,013,240,005,032,210,255,208
65 DATA 244,032,231,255,076,204,255

```

Fig. 31-16.



64 DOS Wedge Disabler

Some 64 programs are incompatible with DOS 5.1, the so-called Wedge program. Such incompatibility is seldom a problem if the Wedge is disabled before those programs are run.

```

10 REM 64 DOS WEDGE DISABLER - SANDER
20 IF PEEK(124)=201 OR PEEK(65534)=23 THEN 60
30 PRINT"[CLEAR,DOWN3]@Q":PRINT"[DOWN2]RUN[HOME]"
40 FOR J= 631 TO 633:POKE J,13:NEXT:POKE 198,3
50 END
60 PRINT"[CLEAR]"
70 REM REST OF PROGRAM GOES HERE

```

Fig. 31-17.

The routine in Fig. 31-17 senses whether the Wedge is active, and disables it if it is. Put it at the start of your incompatible programs, and see how your life improves.



64 DOS Wedge Defaulter

You'll like this one if you use the 64's DOS 5.1 Wedge program on a computer with a Reset button. Under normal circumstances, every time you use the reset button you must reactivate DOS 5.1 by doing a SYS 52224. Then, to re-establish drive 8 as the default device, you must enter @#8 or POKE 186,8.

Figure 31-18 alters the DOS 5.1 program to make that second step unnecessary. Type it in and save it immediately, because it self-destructs when run. To use the program, load it, and put a disk containing DOS 5.1 in drive 8. When you run the program, it will load DOS 5.1, make the necessary changes, then save the changed program as DOS 5.1D (D for default).

```

100 PRINT"[CLEAR,RVS] 64 DOS WEDGE DEFAULTER - FRANK BIONDI[SPACE2]"
110 PRINT"[DOWN]THIS MODIFIES THE 64'S DOS 5.1 PROGRAM"
120 PRINT"TO SET THE DEFAULT DEVICE TO DRIVE 8."
130 PRINT"[DOWN]IT SAVES THE NEW VERSION AS DOS 5.1D."
140 PRINT"[DOWN]PRESS 'STOP' TO ABORT, OR INSERT A DISK"
150 PRINT"CONTAINING DOS 5.1 INTO DRIVE 8, THEN"
160 PRINT"PRESS 'D'....."
170 GET A$:IF A$<>"D" THEN 170
180 A=A+1:IF A=1 THEN LOAD"DOS 5.1",8,1
190 POKE 52225,89:POKE 52226,207
200 FOR J=53081 TO 53087:READ K:POKE J,K:NEXT
210 DATA 169,008,133,186,076,225,204
220 POKE 43,0:POKE 44,204
230 POKE 45,96:POKE 46,207
240 SAVE"DOS 5.1D",8:SYS 64738

```

Fig. 31-18.

Whenever you use DOS 5.1D, SYS 52224 will not only activate the program, but will also set the default device to drive #8. (The default device number is determined by the second data item in line 50. You can change it if you'd like.)

*Frank Biondi
Pittsburgh, Pennsylvania*

Related Tips in Other Chapters

If you like disk utility programs, then Chapter 11 "Floppy Disks" and Chapter 12 "Disk Operations" will also be of interest.

The following specific tips may interest you, as well:

Chapter 12	Double-sided 1571 in 64 Mode 1571 Motor Shutoff
Chapter 24	Screen Saver

Other Utility Programs

*“Utility is the great idol of the age,
to which all powers must do service
and all talents swear allegiance.”*

Friedrich von Schiller
On the Aesthetic Education of Man
1795

Powerful programming talents can serve up some excellent utilities, as the gems in this chapter will demonstrate. They can help you with everything from setting the controls on your monitor, to chaining programs together, to filling memory with zeroes.



Video Test Pattern Generator

The program in Fig. 32-1 is very useful in setting the controls on the video monitor or television display that you use with your computer.

In line 160, words that are not inside brackets should be typed exactly as they appear in the listing. They will become labels for color bars in the test pattern. Also, be especially careful in typing line 180; don't forget the semicolon at its end.

To use the program:

- Run it and press a key as instructed. You should see a pattern of eight vertical bars in different colors, with labels at the top of the screen and an instruction line at the bottom. If you get something else, check your typing.
- If your monitor or TV has an automatic brightness or color control, turn it off at this point. (From now on, we'll call this equipment a "monitor," even if it's usually used for watching football.)

```

100 PRINT"[CLEAR,RVS,SPACE2]VIDEO TEST PATTERN - LOUIS F. SANDER
[SPACE2]"
110 PRINT:PRINT" THIS HELPS YOU OPTIMIZE THE SETTINGS"
120 PRINT" ON YOUR MONITOR OR TELEVISION SET.
130 PRINT"[DOWN] PRESS STOP TO QUIT NOW, OR PRESS ANY
140 PRINT" OTHER KEY TO SEE THE TEST PATTERNS...":GOSUB 270
150 POKE 53280,0:POKE 53281,0: S$="[SPACE5]"
160 PRINT"[WHITE]PURP WHT[SPACE2]BRN[SPACE2]ORNG LRED CYAN RED
[SPACE2]YEL
170 FOR J=1 TO 23
180 PRINT"[RVS,PURPLE]";S$;"[WHITE]";S$;"[BROWN]";S$;"[ORANGE]";S$;
"[L. RED]";S$;"[CYAN]";S$;"[RED]";S$;"[YELLOW]";S$;
190 NEXT
200 PRINT"[RVOFF,WHITE]SET BRIGHT, COLOR, TINT; THEN PRESS KEY";
:GOSUB 270
210 FOR J=1 TO 960:PRINT"[SHFT +]";:NEXT
220 PRINT"SET CONT, HOR & VERT, THEN PRESS A KEY.";:GOSUB 270
230 GOTO 100
240 END
250 :
260 REM WAIT FOR KEYPRESS, CLEAR SCREEN
270 GET A$:IF A$="" THEN 270
280 PRINT"[CLEAR]";
290 RETURN

```

Fig. 32-1.

- Turn the monitor's brightness control all the way up, then turn it back down slowly, stopping as soon as the border appears deep black.
- Adjust your monitor's color control until the colored bars have a pleasing intensity. Don't worry whether they are exactly the right color—just make them colorful enough to please the eye.
 - Use the tint control to give the bars the proper colors.
 - Press any key. If you've typed the program correctly, your screen will be covered with a crosshatch pattern of black squares outlined in white. Set the monitor's Contrast control to give a pleasing overall appearance.
 - Go through all these steps again and touch up the settings of your controls. That ends the use of this program for setting up your monitor.

You can use the crosshatch pattern to make more sophisticated adjustments on the monitor. If the squares aren't the same size in all areas of the screen, they can probably be made so by touching up the vertical height and linearity controls on your monitor. These are often found on the back of the housing or under a flip-up panel on the front.

If the crosshatch isn't uniformly white in all areas of the screen, your monitor's convergence may be out of adjustment. If it's bad enough to be distracting, see a qualified technician to have the adjustment made.



Sprite Finder

Figure 32-2 finds the memory location of unknown sprites, so it's valuable for those who like to use sprites made by others. Once the sprite is in memory, load and run this program. (You may have to use your Reset button if the sprite program disables the Restore key.)

Use the Plus and Minus keys to search through memory for sprites. The Spacebar toggles between high resolution and multicolor sprites, and the Equals key will print the sprite data to the screen.

*Rajesh Batra
Cincinnati, Ohio*



64 Sixpack Plus

The powerful utility in Fig. 32-3 gives life to the 64's "dead" function keys, adds a no scroll feature, and allows quick cursor

```

100 T$="SPRITE FINDER - RAJESH BATRA"
110 B=200:IF PEEK(65534)=23 THEN B=56:POKE 4861,1
    :REM FIRST BLOCK TO SEE
120 NL$=CHR$(141):T$=T$+NL$+"[DOWN]SEARCHES MEMORY FOR SPRITES"
130 T$=T$+NL$+"[DOWN8,RVS]+[RVOFF] AND [RVS]-[RVOFF]
    KEYS CHANGE BLOCK NUMBER."
140 T$=T$+NL$+"[DOWN,RVS]SPACE[RVOFF] TOGGLES HIRES & MULTICOLOR
    MODES."
150 T$=T$+NL$+"[DOWN,RVS]=[RVOFF] KEY DISPLAYS SPRITE DATA."
160 T$=T$+NL$+"[DOWN,SPACE3]- ALL KEYS REPEAT -[DOWN4]"
170 T=0:M$="HIRES":POKE 53281,11:POKE 53280,12:POKE 650,128
180 POKE 53264,0:POKE 53269,1:POKE 53248,100:POKE 53249,96
    :POKE 53287,1
190 POKE 53285,0:POKE 53286,4
200 PRINT"[CLEAR]";T$:PRINT"[DOWN]BLOCK: ";B;TAB(18);"MODE: ";M$
210 POKE 2040,B:POKE 53276,T
220 GET A$:IF A$="" THEN 220
230 IF A$="+" THEN B=B+1:IF B>254 THEN B=254
240 IF A$="-" THEN B=B-1:IF B<1 THEN B=1
250 IF A$=" " THEN 280
260 IF A$="=" THEN POKE 53249,114:GOTO 310
270 GOTO 200
280 IF T=1 THEN T=0:M$="HIRES[SPACE5]":GOTO 200
290 IF T=0 THEN T=1:M$="MULTICOLOR"
300 GOTO 200
310 PRINT:FOR P=0 TO 62:PRINT PEEK(B*64+P);",",":NEXT:GOTO 320
320 PRINT:PRINT"[DOWN]PRESS [RVS]C[RVOFF] TO CONTINUE..."
330 GET A$:IF A$<>"C" THEN 330
340 POKE 53249,96:GOTO 200

```

Fig. 32-2.

movement to the bottom left corner of the screen. It also allows screen color adjustments with easy key combinations.

Written in machine language for speed, "Sixpack Plus" can be used with BASIC and many machine language programs, including the DOS 5.1 wedge utility program.

Programmers can change the ASCII values in lines 70-76 to customize the function key commands. As the program stands, each F-key command can be up to 15 characters long, except for F7. If you add another line of DATA statements and increase the 204 in line 30, F7 can be made to print longer strings.

Here's a guide to the features:

- 1) F1 prints the keyword DATA
- 2) F3 prints LOAD"\$",8

- 3) F5 prints VERIFY"*",8
- 4) F7 prints LIST
- 5) Shift or Shift Lock act as No Scroll keys when using the LIST command. Press to stop listing, release to resume.
- 6) [CTRL DOWN] moves the cursor to the bottom left "home" position.
- 7) SYS 51968 makes the cursor disappear temporarily, letting you use [CMDR B] to change the Border color and [CMDR S] to change the Screen color. Press Return to get your cursor back when you're finished.

```

10 PRINT"[CLEAR,RVS,SPACE3]64 SIXPACK PLUS - KARL JOHNSON III
[SPACE3]"
20 PRINT"[DOWN2]WORKING..."
30 FOR J=51968 TO J+204:READ K:CS=CS+K:POKE J,K:NEXT
40 IF CS<>21875 THEN PRINT"ERROR IN DATASTATEMENTS!":STOP
50 SYS 52005
51 DATA 072,138,072,152,072,008,032,228
52 DATA 255,201,191,208,003,238,032,208
53 DATA 201,174,208,003,238,033,208,201
54 DATA 013,208,235,169,000,133,198,040
55 DATA 104,168,104,170,104,162,058,160
56 DATA 203,142,006,003,140,007,003,120
57 DATA 162,070,142,020,003,140,021,003
58 DATA 088,096,072,173,141,002,201,001
59 DATA 240,249,104,076,026,167,165,197
60 DATA 201,064,240,022,201,007,240,021
61 DATA 165,215,201,133,240,032,201,134
62 DATA 240,028,201,135,240,024,201,136
63 DATA 240,020,076,049,234,166,145,224
64 DATA 251,208,008,162,024,160,000,024
65 DATA 032,240,255,076,049,234,166,154
66 DATA 224,003,208,247,056,233,133,010
67 DATA 010,010,010,024,105,152,133,251
68 DATA 169,203,133,252,160,000,177,251
69 DATA 240,225,032,210,255,200,208,246
70 DATA 068,065,084,065,032,000,000,000
71 DATA 000,000,000,000,000,000,000,000
72 DATA 076,079,065,068,034,036,034,044
73 DATA 056,000,000,000,000,000,000,000
74 DATA 086,069,082,073,070,089,034,042
75 DATA 034,044,056,000,000,000,000,000
76 DATA 076,073,083,084,000

```

Fig. 32-3.

Simultaneously pressing Stop and Restore will disable Sixpack Plus; it can be reactivated by executing SYS 51968 or SYS 52005.

*Karl Johnson III
Houston, Texas*



64 Datamaker

Tired of entering DATA statements? Or would you like another way to compact data in your programs? Use one of the Datamaker programs in Figs. 32-4 and 32-5. These programs not only create DATA statements for you, they also create the subroutine to load them!

Caution: When typing the programs, be aware that each has two lines that will exceed the 80-character limit unless you handle them in a special way. In either program's line 63030, you must eliminate all spaces between keywords. And in the last line of either program, you must use question marks to abbreviate both of the PRINT statements.

Once you've typed them in, just run either program and enter the starting and ending addresses of the data you want to convert, along with the line number of the first DATA statement that will ultimately appear in your program. Datamaker creates the data and enters the copy routines, and then asks you if you want to delete Datamaker. At this point, you can simultaneously press Stop and Restore to stop, or press Return to delete Datamaker's program lines. (Ignore the UNDEFINED LINE NUMBER error.)

The difference between the two Datamaker programs is the form of data they create. In DATAMAKER.DEC, the data is stored as the usual decimal numbers—bulky, but easy to work with. DATAMAKER.ASC, on the other hand, converts each value into a two letter code, which stores more compactly in the program. Otherwise, they both do identical jobs.

*David Pankhurst
Montreal, Quebec
Canada*



64 Timer

The program in Fig. 32-6 uses what we call “interrupts” to display the time in hh:mm:ss format in the top right corner of the screen. The 64's seldom-used Time Of Day (TOD) clock keeps track of the time, instead of the commoner but less accurate TI\$ variable.

What all this means is that you can have a time display, updated 60 times each second, while working or running another program. The interrupts are used only to display the time, not to keep track of it.

Chapter 32

```

63000 PRINT"[CLEAR,RVS,SPACE3]64 DATAMAKER.DEC - DAVID PANKHURST
[SPACE3]"
63010 PRINT"[DOWN]CONVERTS MEMORY INTO DATA STATEMENTS.[DOWN2]"
63020 INPUT"MEMORY START, END, AND 1ST LINE NUMBER";AS,AE,LS:X=63040
63030 X$="RE[SHFT S]:F[SHFT O]I=0TO-1ST[SHFT E]-1:R[SHFT E]X$
:I=V[SHFT A](X$)=-1:N[SHFT E]:X="+STR$(AS)+" :D[SHFT A]-1"
:GOTO 63090
63040 X$="R[SHFT E]Y:IFY<>999T[SHFT H]P[SHFT O]X,Y:X=X+1
:G[SHFT O]" +STR$(LS):X=63050:GOTO 63090
63050 X$="RE[SHFT T]":X=63060:GOTO 63090
63060 FOR I=0 TO 6
63070 X$=X$+RIGHT$(STR$(1E7+PEEK(AS)),3)+",":AS=AS+1:I=I-(AS>AE)*9
63080 NEXT:X$="D[SHFT A]" +LEFT$(X$,LEN(X$)-1):X=63110
63090 PRINT"[CLEAR,DOWN2]";LS;X$:PRINT"AS=";AS;":AE=";AE;
:LS=";LS+10;":G[SHFT O]";X; "[HOME]";
63100 POKE 631,13:POKE 632,13:POKE 198,2:END
63110 IF AS<=AE THEN 63060
63120 X$="D[SHFT A] 999":X=63130:GOTO 63090
63130 X=63000:INPUT"READY TO DELETE";X$
63140 PRINT"[CLEAR,DOWN2]";X:PRINT"X=";X+10;":G[SHFT O]63140[HOME]";
:POKE 631,13:POKE 632,13:POKE 198,2:END

```

Fig. 32-4.

```

63000 PRINT"[CLEAR,RVS,SPACE3]64 DATAMAKER.ASC - DAVID PANKHURST
[SPACE3]"
63010 PRINT"[DOWN]CONVERTS MEMORY INTO DATA STATEMENTS.[DOWN2]"
63020 INPUT"MEMORY START, END, AND 1ST LINE NUMBER";AS,AE,LS:X=63040
63030 X$="RE[SHFT S]:F[SHFT O]I=0TO-1ST[SHFT E]-1:R[SHFT E]X$
:I=V[SHFT A](X$)=-1:N[SHFT E]:X="+STR$(AS)+" :D[SHFT A]-1"
:GOTO 63110
63040 Y$=CHR$(34):X$="R[SHFT E]X$:IFX$="+Y$+"/E"+Y$+"T[SHFT H]RE
[SHFT T]":X=63050:GOTO 63110
63050 X$="F[SHFT O]I=1TOLEN(X$)ST[SHFT E]2
:P[SHFT O]X,A[SHFT S](M[SHFT I](X$,I,1))*16+A[SHFT S](M
[SHFT I](X$,I+1,1))-1105"
63060 X=63070:GOTO 63110
63070 X$="X=X+1:NEXT:G[SHFT O]" +STR$(LS-20):X=63080:GOTO 63110
63080 FOR I=0 TO 13:X=PEEK(AS):Y=INT(X/16)
63090 X$=X$+CHR$(Y+65)+CHR$(X-Y*16+65):AS=AS+1:I=I-(AS>AE)*99:NEXT
63100 X$="D[SHFT A]" +X$:X=63130
63110 PRINT"[CLEAR,DOWN2]";LS;X$:PRINT"AS=";AS;":AE="AE":LS="LS+10"
:G[SHFT O]";X; "[HOME]";
63120 POKE 631,13:POKE 632,13:POKE 198,2:END
63130 IF AS<=AE THEN 63080
63140 X$="D[SHFT A] /E":X=63150:GOTO 63110
63150 X=63000:INPUT"READY TO DELETE";X$
63160 PRINT"[CLEAR,DOWN2]"X:PRINT"X="X+10;":G[SHFT O]63160[HOME]";
:POKE 631,13:POKE 632,13:POKE 198,2:END

```

Fig. 32-5.

```

10 PRINT"[CLEAR,RVS] 64 TIMER - STEPHANE ST-LOUIS[SPACE11]"
11 DATA 120,169,019,141,020,003,169,192
12 DATA 141,021,003,088,096,041,015,024
13 DATA 105,048,096,173,011,220,032,013
14 DATA 192,141,032,004,173,010,220,032
15 DATA 013,192,141,035,004,173,009,220
16 DATA 032,013,192,141,038,004,173,011
17 DATA 220,032,110,192,041,001,032,015
18 DATA 192,141,031,004,173,010,220,032
19 DATA 110,192,041,007,032,015,192,141
20 DATA 034,004,173,009,220,032,110,192
21 DATA 041,007,032,015,192,141,037,004
22 DATA 169,032,141,030,004,141,039,004
23 DATA 169,058,141,033,004,141,036,004
24 DATA 173,008,220,076,049,234,074,074
25 DATA 074,074,096
100 PRINT"[DOWN]PUTS A REAL-TIME DISPLAY ON THE SCREEN."
110 PRINT"[DOWN]STOP/RESTORE KILLS, SYS 49152 REENABLES."
120 FOR J=49152 TO 49266:READ K:POKE J,K:CS=CS+K:NEXT
130 IF CS<>9872 THEN PRINT"DATA ERROR":END
140 INPUT"[DOWN,SPACE2]HOUR";H$:IF LEN(H$)>1 THEN H=16
150 HH=VAL(RIGHT$(H$,1)):H=H+HH:POKE 56331,H
160 INPUT"MINUTE";M$:IF LEN(M$)>1 THEN M=16*VAL(LEFT$(M$,1))
170 MM=VAL(RIGHT$(M$,1)):M=M+MM:POKE 56330,M
180 POKE 56329,0:POKE 56328,0
190 SYS 49152

```

Fig. 32-6.

Stop/Restore deactivates the display, but executing a SYS 49152 will reenables it. Even with the display deactivated, the clock just keeps on ticking!

*Stephane St-Louis
Ste-Therese, Quebec
Canada*



64 Variable Dump

The program in Fig. 32-7 lets you see the values of all variables that exist in system memory. It can be an invaluable aid to debugging BASIC programs.

To use the program, type it in, save it to disk, then run it. If you get an error message, recheck your DATA statements. Once the program has been run successfully, entering the SYS command shown on the screen will give you a dump of all the variables in memory. Pressing the Stop key will pause the listing for the duration of the keypress. Pressing the Control key will slow the listing down.

Chapter 32

```
10 PRINT"[CLEAR,RVS,SPACE4]64 VARIABLE DUMP - SHACHAR EBEL[SPACE5]"
15 PRINT"[DOWN]THIS DISPLAYS THE VALUES OF ALL"
20 PRINT"VARIABLES IN A 64 BASIC PROGRAM."
25 PRINT"[DOWN]WORKING[UP]"
30 SA=49152:REM RELOCATABLE START ADDR
35 FOR J=1 TO 193:READ K:CS=CS+K:NEXT
40 IF CS<>24349 THEN PRINT"DATA ERROR":STOP
45 RESTORE:FOR J=SA TO SA+188:READ K:POKE J,K:NEXT
50 AD=SA+47:GOSUB 60:AD=SA+170:FOR J=0 TO 2:GOSUB 60:NEXT
55 PRINT"SYS";SA;"WILL DUMP THE VARIABLES.":END
60 READ K:POKE SA+K,AD-256*INT(AD/256):POKE SA+K+1,INT(AD/256)
  :RETURN
61 DATA 165,045,164,046,133,251,132,252
62 DATA 160,001,185,251,000,217,047,000
63 DATA 208,004,136,016,245,096,032,047
64 DATA 192,024,165,251,105,007,133,251
65 DATA 144,002,230,252,169,013,032,022
66 DATA 231,165,145,016,252,048,217,160
67 DATA 000,177,251,170,041,127,032,022
68 DATA 231,200,177,251,072,041,127,168
69 DATA 032,022,231,104,016,063,138,048
70 DATA 041,169,036,032,170,192,169,034
71 DATA 032,022,231,177,251,153,251,000
72 DATA 200,192,005,208,246,165,253,240
73 DATA 012,160,000,177,254,032,022,231
74 DATA 200,196,253,208,246,169,034,076
75 DATA 022,231,169,037,032,170,192,177
76 DATA 251,170,200,177,251,168,138,032
77 DATA 145,179,024,144,031,169,032,032
78 DATA 170,192,177,251,153,095,000,200
79 DATA 192,007,208,246,165,098,170,009
80 DATA 128,133,098,160,000,138,016,002
81 DATA 160,255,132,102,032,221,189,076
82 DATA 030,171,032,022,231,152,208,005
83 DATA 169,032,032,022,231,169,061,032
84 DATA 022,231,160,002,096,023,076,117
85 DATA 136
```

Fig. 32-7.

The program places 189 bytes of machine language into memory. Variable SA in line 30 sets the starting address of this code, which can be anywhere there are 189 free bytes. Memory location 820 is another good place to put it. Wherever you put the machine language, "64 Variable Dump" will tell you the proper SYS command.

Shachar Ebel
Rehovot, Israel

**64 Pauser**

Type Fig. 32-8 and run it, and you'll have a convenient way to pause most BASIC and many machine language programs, not to mention listings to the screen.

After you've run the program, load another one that you'd like to experiment with. When you're ready for a pause, just hold down the left Shift key (or the Shift Lock key), then tap the Control key and everything will stop. The screen border will turn light grey to alert you to the paused condition. Releasing the Shift or Shift Lock key will resume execution.

Because the Control/Left Shift combination is seldom used in normal operation, there's very little chance of your inadvertently pausing your main program.

Using the Stop/Restore key combination will remove your ability to pause programs. To reactivate it, enter **SYS 679**.

After you've saved this program and are familiar with how it works, you might want to add a **NEW** statement at the end of line 70. This will remove the BASIC loader once the utility is active.

*Jason D. Clark
Las Vegas, Nevada*

```

10 PRINT"[CLEAR,RVS,SPACE9]64 PAUSER - JASON CLARK[SPACE8]"
20 PRINT"[DOWN]TO PAUSE A PROGRAM OR LISTING, HOLD DOWN"
30 PRINT"[UP]LEFT SHIFT OR SHIFT LOCK, THEN HIT CTRL."
40 PRINT"RELEASE THE SHIFT TO END THE PAUSE."
50 PRINT"[DOWN]STOP/RESTORE DISABLES THE FEATURE,"
60 PRINT"SYS 679 RE-ENABLES IT.[DOWN]"
70 FOR J=679 TO 746:READ X:POKE J,X:NEXT:SYS 679
71 DATA 173,020,003,141,202,002,173,021
72 DATA 003,141,203,002,120,169,002,141
73 DATA 021,003,169,192,141,020,003,088
74 DATA 096,173,141,002,041,005,201,005
75 DATA 240,003,076,049,234,238,032,208
76 DATA 169,253,141,000,220,173,001,220
77 DATA 041,128,240,244,169,255,141,000
78 DATA 220,206,032,208,169,000,141,141
79 DATA 002,076,201,002

```

Fig. 32-8.

**64 Remvert**

The little utility in Fig. 32-9 can highlight **REM** statements by putting their text in reverse field. Once a program has been

“highlighted” the effect is permanent; the reversed REMs will even appear on most printer listings. I’ve found this effect very helpful in designing and editing my BASIC programs.

The program is written in BASIC, and POKEs machine code into memory. It is fully relocatable to any free area of memory. Because it works by placing an RVS code in the first position after the REM itself, it’s a good idea to put an extra space there before doing the highlighting. Otherwise, you may lose the first character of your remark.

Mark Rea
Address Unknown

```

10 PRINT"[CLEAR,RVS,SPACE9]64 REMVERT - MARK REA[SPACE10]"
20 SA=53000:REM  RELOCATABLE START ADDR
30 FOR J=SA TO SA+62:READ K:POKE J,K:CS=CS+K:NEXT
40 IF CS<>9980 THEN PRINT "ERROR IN DATA STATEMENTS":END
50 PRINT"[DOWN]TO MAKE A PROGRAM'S REMS LIST IN [RVS]RVS[RVOFF] ,"
60 PRINT"JUST LOAD THE PROGRAM AND SYS";SA;"[LEFT]. "
70 PRINT"[DOWN]IT'S BEST TO FIRST INSERT AN EXTRA"
80 PRINT"SPACE AFTER EACH REM KEYWORD.[DOWN]"
90 DATA 165,043,133,251,165,044,133,252
91 DATA 160,003,200,177,251,240,031,201
92 DATA 143,240,002,208,245,200,177,251
93 DATA 240,020,201,032,208,012,200,177
94 DATA 251,201,032,240,249,201,018,240
95 DATA 001,136,169,018,145,251,160,001
96 DATA 177,251,170,136,177,251,133,251
97 DATA 134,252,224,000,208,202,096

```

Fig. 32-9.



64 Quick Erase

The routine in Fig. 32-10 puts zeroes into all the 64’s RAM between 2048 and 65535, then resets the computer. It can be used as an exit routine which will make sure that your program code doesn’t stay behind as a target for nosy hackers.

After lines 20 to 60 have been executed, entering a SYS 828 will zero out all memory. Be sure to save the program before you try it. (Residing in memory itself, it will be clobbered by the SYS 828.)

Ranjan Bose
Winnipeg, Manitoba
Canada

```

10 REM 64 QUICK ERASE - RANJAN BOSE
11 DATA 169,000,133,251,169,008,133,252
12 DATA 169,000,168,145,251,200,196,251
13 DATA 208,249,230,252,169,250,197,252
14 DATA 208,238,032,226,252,096
20 FOR J=828 TO 857
30 READ K
40 CS=CS+K
50 POKE J,K
60 NEXT
70 IF CS<>5354 THEN PRINT "ERROR IN DATA STATEMENTS":STOP
80 PRINT"SYS828 WILL ERASE ALL MEMORY!"

```

Fig. 32-10.



64 Search

When writing long programs, you can spend an inordinate amount of time sifting through listings to find certain variables or subroutine calls. The program in Fig. 32-11 can be used to locate any desired variable, string or BASIC keyword in a 64 BASIC program.

Each line containing the searched-for item will be displayed on the screen. Screen output can be redirected to the printer via the CMD command.

To use the utility, just run the attached program. Then load the BASIC program you want to search through, and enter the search command in the following format:

```
SYS 49152,searchitem
```

The *searchitem* can be anything at all in the program, and BASIC commands may be abbreviated if you'd like. Don't use quotation marks unless you want to search for them.

The utility's usefulness becomes apparent when you are working with a long program. For example, I was recently working with a 1,056-line program and I had to change the starting line number of a subroutine. Without this utility, I would have had to sift through every one of the 1,055 other lines to find where calls to this subroutine were made. Instead, I just used the search command with the subroutine's old line number as *searchitem*.

Working with the utility is easier if you use a memory saver program to save the machine language directly, then reload it whenever it's needed. Its start address is 49152, and its end address is 49354.

Michael J. Gibbons
Orange City, Florida

```

10 PRINT"[CLEAR,RVS,SPACE6]64 SEARCH - MICHAEL GIBBONS[SPACE7]"
20 PRINT"[DOWN]WORKING...":FOR J=1 TO 203:READ K:POKE 49151+J,K
   :CS=CS+K:NEXT
30 IF CS<>30072 THEN PRINT"ERROR IN DATA STATEMENTS!":STOP
40 PRINT"[UP]TO SEARCH FOR XXX, ENTER SYS49152,XXX":PRINT:PRINT
41 DATA 160,000,200,177,122,153,205,192
42 DATA 208,248,160,000,152,153,000,002
43 DATA 200,192,080,208,248,169,008,133
44 DATA 252,169,001,133,251,160,002,177
45 DATA 251,141,203,192,200,177,251,141
46 DATA 204,192,200,177,251,240,027,205
47 DATA 206,192,208,246,140,205,192,162
48 DATA 000,232,200,189,206,192,240,028
49 DATA 209,251,240,245,172,205,192,076
50 DATA 042,192,200,024,152,101,251,133
51 DATA 251,144,002,230,252,160,001,177
52 DATA 251,208,194,096,173,203,192,133
53 DATA 020,173,204,192,133,021,032,019
54 DATA 166,160,001,132,015,032,215,170
55 DATA 160,003,132,073,173,204,192,174
56 DATA 203,192,032,205,189,169,032,164
57 DATA 073,041,127,032,071,171,201,034
58 DATA 208,006,165,015,073,255,133,015
59 DATA 200,240,004,177,095,208,011,165
60 DATA 096,133,252,165,095,133,251,076
61 DATA 074,192,016,223,201,255,240,219
62 DATA 036,015,048,215,056,233,127,170
63 DATA 132,073,160,255,202,240,008,200
64 DATA 185,158,160,016,250,048,245,200
65 DATA 185,158,160,048,186,032,071,171
66 DATA 208,245,096

```

Fig. 32-11.



64 BASIC Linker

The ability for one BASIC program to load and run another can be very useful, but it can be somewhat tricky to do. If you merely use the LOAD command from within a program, you can easily get into trouble.

First of all, a program mode LOAD does not initialize the variables of the loaded program. Any variable that was used in the first program will retain its value when the second program starts running. Because most programs rely on all variables being zeroed when a run is started, many troubles can result.

Connected with retaining the value of the variables is the requirement that the loading program is longer than the loaded one. If

this requirement is ignored, the loaded program will corrupt itself and fail to run properly.

Figure 32-12 creates a machine language routine that allows one BASIC program to load and run another, without having to worry about variable values or program lengths.

After the machine language is poked into memory, your BASIC program can load and run another BASIC program by executing a statement of the form:

```
SYS 49152,"programname",8
```

where "programname" is the name of the program you want to load.

That's all there is to it. Just don't forget to include the drive number at the end: if you want the program to load from another drive, substitute its number for the 8 shown in the example.

The machine language is totally relocatable, so if you want to put it at a different starting address, just change the value of SA in line 20. Also, you may want to save the machine language itself, to avoid having to poke it into memory every time you use it. With the starting address as shown, it occupies memory from 49152 - 49208.

Michael J. Gibbons
Orange City, Florida

```
10 PRINT"[CLEAR,RVS,SPACE3]64 BASIC LINKER - MICHAEL GIBBONS[SPACE4]
"
20 SA=49152 :REM RELOCATABLE START ADDR
30 PRINT "[DOWN2]WORKING...":FOR J=0 TO 56:READ K:POKE SA+J,K
:CS=CS+K:NEXT
40 IF CS<>6644 THEN PRINT"ERROR IN DATA STATEMENTS!":STOP
50 C$="[LEFT]","Q$=CHR$(34):PRINT"[UP]TO HAVE ONE BASIC PROGRAM
LOAD AND RUN"
60 PRINT:PRINT"ANOTHER, HAVE THE FIRST PROGRAM EXECUTE:"
70 PRINT"[DOWN,SPACE5]SYS";SA;C$;Q$;"PROGRAMNAME";Q$;"DRIVE#[DOWN2]
"
71 DATA 032,253,174,169,000,133,010,032
72 DATA 212,225,165,010,166,043,164,044
73 DATA 032,213,255,176,033,032,066,166
74 DATA 032,183,255,041,191,240,005,162
75 DATA 029,076,055,164,134,045,134,047
76 DATA 132,046,132,048,169,000,032,144
77 DATA 255,104,104,076,089,166,076,249
78 DATA 224
```

Fig. 32-12.



64 Machine Language Loader

Related to the problem of having a BASIC program load another BASIC program is the problem of having a BASIC program load a machine language program.

Instead of using the usual cumbersome code, you might want to use the program shown in Fig. 32-13.

After its machine language is poked into memory, the following statement can be used anywhere in a BASIC program to load a machine language file:

```
SYS 820,"filename"
```

where "filename" is the name of the machine language file you wish to load.

Notice that you don't include the drive number.

What we've called "machine language" can, of course, be anything that fills a specific range of memory — a sprite, a group of sprites, a hi-res screen, and so forth.

The machine language from our program is completely relocatable, so if you want it to reside in another area of memory, change the value of SA in line 30.

As with the other programs above, you may want to save this one's machine language by itself, to avoid having to poke it into memory every time you want to use it. You can use one of this book's memory saver programs to accomplish the task.

*Michael J. Gibbons
Orange City, Florida*

```
10 PRINT"[CLEAR,RVS,SPACE2]64 MACHINE LANGUAGE LOADER - GIBBONS
[SPACE2]"
20 :
30 SA=820:REM RELOCATABLE START ADDRESS
40 :
50 FOR J=0 TO 34:READ K:POKE SA+J,K:CS=CS+K:NEXT
60 IF CS<>4298 THEN PRINT"ERROR IN DATA STATEMENTS!":STOP
70 C$="[LEFT]",":Q$=CHR$(34):PRINT"[DOWN]
  TO HAVE A BASIC PROGRAM LOAD ML FROM"
80 PRINT"[DOWN] DRIVE #8, HAVE IT EXECUTE:"
90 PRINT"[DOWN2,SPACE10]SYS";SA;C$;Q$;"MLFILENAME";Q$;"[DOWN2]"
91 DATA 032,253,174,169,001,162,008,168
92 DATA 032,186,255,032,158,173,169,013
93 DATA 208,005,162,022,076,055,164,032
94 DATA 166,182,032,189,255,169,000,032
95 DATA 213,255,096
```

Fig. 32-13.

△ 64 Appender

For those who have wished for a simple way to combine two or more programs into one, the program in Fig. 32-14 will do the trick. Type it in and save it immediately, because it self-destructs when run. After you run it, load your first program, type SYS 700 and press Return. Then load your second program. If you want to chain more together, enter SYS 700 again after loading each one.

When you've loaded the last program, enter SYS 723, and you'll see the combined programs in memory as one long program, even if some of them have duplicated line numbers. If there are duplications, or if the numbers run any way but from low to high, you'll have problems when you try to run or edit the combined programs. (If you watch the results of your editing attempts when that happens, you can learn a lot about how BASIC performs its various editing tasks.)

You can avoid problems by renumbering the component programs before you combine them. Usually, it's all right to do this while they are in memory, before you execute your SYS 700. When renumbering, make sure the lowest number in each program is greater than the highest number in the preceding one.

Like all appender programs, this one fiddles with BASIC's inner workings, and is prone to unusual failures. It doesn't seem to work, for example, with any program that starts with line number 0. Be alert for these failures every time you use the program.

David L. Prestemon
Derwood, Maryland

```

100 PRINT"[CLEAR,RVS,SPACE6]64 APPENDER - D.L. PRESTEMON[SPACE6]"
110 PRINT"[DOWN]TO CHAIN 64 PROGRAMS TOGETHER:"
120 PRINT"[DOWN]1. LOAD OR TYPE THE FIRST PROGRAM"
130 PRINT"2. ENTER SYS 700"
140 PRINT"3. LOAD OR TYPE THE NEXT ONE"
150 PRINT"4. REPEAT 2 & 3 AS DESIRED"
160 PRINT"5. WHEN DONE, ENTER SYS 723[DOWN]"
170 FOR J=700 TO 731:READ K:POKE J,K:CS=CS+K:NEXT
180 IF CS<>3904 THEN PRINT "ERROR IN DATA!":STOP
190 DATA 160,000,230,043,208,002,230,044
200 DATA 177,043,208,246,230,043,208,002
210 DATA 230,044,177,043,208,236,096,169
220 DATA 001,133,043,169,008,133,044,096
230 NEW

```

Fig. 32-14.

 ☆ 128 Appender

Building up programs from subroutines is a lot more convenient if you can easily append one program to another. Figure 32-15's short BASIC loader puts a machine language routine into the 128's cassette buffer to make appending as easy as pressing a few keys.

Run this program to enable "128 Appender". Then do a NEW to remove it from memory. Now load a program to which you want to append one or more others. To append a program to the one already in memory, just press the F1 key and load the second program in the normal way. When the READY prompt returns, press the F3 key. If you do a LIST at this point, you'll see that the program in memory consists of your first one with your second appended to it. You can continue to append others using the same routine—press F1, load a program, then press F3.

Because "128 Appender" doesn't look at line numbers, it's up to you to be sure that all line numbers in the appended program are higher than the highest line number in the one to which it's appended. The 128's RENUMBER command should make this job a cinch.

Like all appender programs, this one manipulates BASIC's inner workings, and is prone to unusual failures. For example, it doesn't seem to work with any program that starts with line number 0. Be alert for these failures every time you use it.

Stanley L. Anderson
Tishomingo, Oklahoma

```

100 PRINT"[CLEAR,RVS,SPACE3]128 APPENDER - STANLEY L. ANDERSON
      [SPACE3]"
110 FOR J=1 TO 48:READ K$:CS=CS+DEC(K$):NEXT
120 IF CS<>2816 THEN PRINT"DATA ERROR":STOP
130 RESTORE:FOR J=2816 TO 2863:READ K$:POKE J,DEC(K$):NEXT
140 KEY 1,"SYS2816:REM LOAD PROGRAM, THEN PRESS F3."+CHR$(13)
150 KEY 3,"SYS2848:REM APPEND COMPLETE!"+CHR$(13)
160 PRINT"[DOWN]TO APPEND A PROGRAM, PRESS F1 THEN LOAD"
170 PRINT"IT. WHEN THE LOAD IS FINISHED, PRESS F3."
180 PRINT"[DOWN]REPEAT TO APPEND ADDITIONAL PROGRAMS.[DOWN]"
190 PRINT"SAVE TO DISK WHEN FINISHED.[DOWN2]"
200 DATA A5,2D,8D,18,0B,A5,2E,8D,19,0B
210 DATA 38,AD,10,12,E9,02,85,2D,AD,11
220 DATA 12,85,2E,60,01,01,01,01,01,01
230 DATA 01,01,AD,18,0B,85,2D,AD,19,0B
240 DATA 85,2E,60,01,01,01,01,01

```

Fig. 32-15.

☆ 128 Mode Guard

Because the 128 offers both 40-column and 80-column display modes, plus the old standby 64 mode, it can be hard to remember just which mode a program is intended for.

The routine in Fig. 32-16, designed to be used as the first few lines of a larger program, can be tailored for the mode in which the larger program runs. If you attempt to use an inappropriate mode, you'll get a warning and on-screen instructions. Where possible, the program sets up the correct mode.

Gordon A. Alexander
Newington, Connecticut

```

0 REM 128 MODE GUARD - G.A.ALEXANDER
1 SN=PEEK(215):REM SN=SCREEN;PM=PROGRAM: ADD TO TOP OF PROGRAMS
2 PM=0 :REM PM=0 FOR 128-40C;128 FOR128-80C;13 FOR 64:CHG TO SUIT
3 RB$=CHR$(7):IF SN=PM THEN 10
4 IF PM=13 AND SN<>13 THEN PRINT RB$;"[CLEAR,DOWN]
64 PROG-RELOAD WHEN 64 SCREEN APPEARS":SLEEP 5:GO 64
5 IF PM<>13 AND SN=13 THEN PRINT"[CLEAR,DOWN]
128 PROG-SET 40/80 BUTTON,RESET 128, RELOAD":END
6 IF PM=0 AND SN=128 THEN PRINT RB$"[CLEAR,DOWN]
SWITCH MONITOR TO 40COL":SLEEP 5:SYS 65375:GOTO 10
8 IF PM=128 AND SN=0 THEN PRINT RB$"[CLEAR,DOWN]
SWITCH MONITOR TO 80COL":SLEEP 5:SYS 65375:GOTO 10
9 GOTO 3
10 PRINT"[DOWN]START YOUR PROGRAM ON LINE 10"

```

Fig. 32-16.

☆ 128 Hi-Res Helper

Using a 128's GRAPHIC mode is great, but it can be a pain to figure out the numbers corresponding to various screen locations. Figure 32-17's little program makes that work a lot easier, by letting you move a tiny pixel cursor around the screen while it reports its own location.

Just load and run the program, then use the numeric keypad to move the little cursor. The 8 key moves it up, the 2 moves it down, and so on. To leave a dot in the cursor's current position, press the 5 key. Press the Stop key to exit the program.

If you delete line 5010, you can use the program in conjunction with other graphics programs you are developing. Just insert "128 Hi-Res Helper" (less line 5010) after the other program's DRAW statements.

```

5000 REM 128 HI-RES HELPER - J. JARVIS
5001 REM USE KEYPAD TO MOVE TINY CURSOR
5002 REM USE STOP KEY TO QUIT
5010 GRAPHIC 1,1:TRAP 5180
5020 X=160:Y=100:GOTO 5160
5030 DRAW 1,X,Y
5040 GET KEY J$:J=VAL(J$)
5050 IF J=5 THEN F=1
5060 IF F=0 THEN DRAW 0,X,Y
5070 IF J=1 THEN Y=Y+1:X=X-1
5080 IF J=2 THEN Y=Y+1
5090 IF J=3 THEN Y=Y+1:X=X+1
5100 IF J=4 THEN X=X-1
5110 IF J=6 THEN X=X+1
5120 IF J=7 THEN Y=Y-1:X=X-1
5130 IF J=8 THEN Y=Y-1
5140 IF J=9 THEN Y=Y-1:X=X+1
5150 IF J<>5 THEN F=0
5160 CHAR 1,1,1,"X="+STR$(X)+" Y="+STR$(Y)+"[SPACE3]"
5170 GOTO 5030
5180 GRAPHIC 0:END

```

Fig. 32-17.

You'll have the little cursor plus whatever your program has drawn.

*Jerry W. Jarvis
Spokane, Washington*

☆ 128 Restore Colors

If you have a 128 and are tired of its default color combinations, then Fig. 32-18 is for you. It sets the colors that appear when you press the Stop/Restore combination, and it also prints your own custom message at the top of the screen!

Just type the program, save it to disk, then run it. If everything works correctly, you will be prompted for the default border color, screen color and cursor color. You'll then be asked to type in the message that will appear at the top of the screen.

When entering the colors, use the numbers from 0-15 that you POKE^o into 53280 and 53281 to change your border and screen.

*David C. May
Address Unknown*

```

100 PRINT"[CLEAR,RVS,SPACE3]128 RESTORE COLORS - DAVID C. MAY
      [SPACE4]"
110 PRINT"[DOWN]CHANGES COLORS PRODUCED BY STOP/RESTORE.":PRINT
120 FOR J=5120 TO 5175:READ K:CS=CS+K:POKE J,K:NEXT
130 IF CS<>6008 THEN PRINT"ERROR IN DATA!"
140 INPUT"BORDER COLOR";BC:INPUT"SCREEN COLOR";SC
      :INPUT"CURSOR COLOR";CC
150 POKE 5146,BC:POKE 5141,SC:POKE 5151,CC:POKE 2560,0:POKE 2561,20
      :PRINT
160 PRINT"[DOWN]MESSAGE TO APPEAR AFTER STOP/RESET:"
170 B$="":FOR I=0 TO 39:DO:GET KEY A$:LOOP WHILE(A$<" " OR A$>"
      [BACK ARROW]")AND A$<>CHR$(13):B$=B$+A$:PRINT A$;
      :IF A$=CHR$(13) THEN LG=I:I=39:ELSE NEXT
180 PRINT:IF I=40 THEN B$=B$+CHR$(13):LG=I
190 POKE 5164,LG+1:FOR I=0 TO LG:POKE 5175+I,ASC(MID$(B$,I+1,1))
      :NEXT:PRINT"DEFAULT COLORS CHANGED!"
200 DATA 032,204,255,032,122,065,032,141
210 DATA 065,032,018,065,032,056,082,169
220 DATA 000,133,021,088,169,000,141,033
230 DATA 208,169,000,141,032,208,169,000
240 DATA 133,241,162,000,189,055,020,032
250 DATA 210,255,232,224,001,208,245,169
260 DATA 013,032,210,255,076,055,077,000

```

Fig. 32-18.

Related Tips in Other Chapters

You might find value in these other utility programs:

Chapter 5	64 Restore/Reset Disable
	64 Delete Right
	64 F7 Apostrophe Key
	64 128 F-Key Conversions
Chapter 13	Quantum Link Password
Chapter 32	Decimal to Binary Converter
	Binary to Decimal Converter
	Decimal to Hex Converter
	Hex to Decimal Converter
	Universal Base Converter
	128 Arabic to Roman Conversion

Commercial Software: General Tips

“Glittering generalities.”

Franklin J. Dickman
Letter, The Providence Journal
December 14, 1849

As Commodore owners, we have access to one of the world's largest collections of commercial software programs. Sometimes it seems that every store in every mall has its rack of Commodore software.

Whether your taste runs to games, productivity software or educational tools, this chapter can help you get more from the programs you buy.



Save Your Receipts

Whenever I buy a new piece of commercial software, I keep the plastic cover and sales receipt. I put them in an envelope marked with the name of the program they belong to. Then if I need to contact the manufacturer, I have all the information about my purchase right there in one handy place. They often ask for a copy of the original sales slip.

*Rodney Weese
Hanover, Pennsylvania*



Software Documentation

If the documentation on your commercial software seems poor or completely lacking, check the directory on the disk. Sometimes manufacturers put the documentation directly on the disk in a readable file. To investigate this, load the directory and look for files with SEQ after the filename.

These files are sequential files and can be read with many word processors or with a sequential file reader program. Some simple SEQ file readers are included in this book; another is the SPRINT program provided on the Quantum Link disk.

Often the filename will indicate just what the SEQ file contains.

*Rebecca Begley
Louisville, Kentucky*



Commercial Software and Your Printer

Finding programs that are compatible with your printer can seem like a real chore, because many programs like “The Print Shop” and “Jingle Disk” list only a handful of compatible printers. But just because your printer isn’t listed doesn’t mean it won’t work.

I own a Seikosha SP-110-VC which works fine with many programs that don’t list it in their instructions. When in doubt, I buy the program, because one of its many printer combinations is bound to work with mine.

*Neal Pynenberg
Kaukauna, Wisconsin*



POKEs and Commercial Software

Some tips for commercial programs suggest making POKEs before the program is run. Often this can give you extra or unlimited lives, or endow you with superhuman powers.

Some people are confused about these POKEs, but the difficulty goes away if you think about what you are doing: First you load the program into memory. Then, before running it, you make the POKEs to change the contents of certain memory locations. Finally, you run the program by typing RUN or a SYS statement and pressing Return.

If you mistakenly perform the POKEs before loading the program, the subsequent LOAD will overwrite the changed memory locations.

But some programs begin executing as soon as they are loaded. What do we do about them? The usual solution is to press your Reset button, make your POKEs, then execute a RUN or a SYS. Basically, the Reset button lets you regain control of your computer while leaving memory intact. Chapter 16 contains more detailed explanations of its functions.



Commercial Software and the Reset Button

If you have installed a Reset button on your computer, you can sometimes use it to start commercial programs. On Pole Position and some others that are started by a SYS command, you need not remember or type the SYS. Just press the Reset button, and the game will start in the usual way.

*Javier Arano
Capital Federal, Argentina*



128 Keys in 64 Mode

If you're using your 128 in 64 mode with commercial software, try all the 128 keys that are normally dead in 64 mode. With Fleet System 2, for example, the No Scroll key acts like Run/Stop and switches to the menu. Other keys do different things, and on some software the numeric keypad may even be enabled.

Regardless of your software, the left mouse button or joystick fire button corresponds to Function Key 8 when the device is plugged into Port 1. Joystick down corresponds to pressing the Insert key, joystick right is equivalent to a Cursor Left, and joystick left is the same as pressing Return.

*Sean Burke
Burton, Ohio*



Using an Extra Hand

Many games in which the action is non-stop require you to press the Spacebar at certain times, such as when you want to throw a hand

grenade. Unfortunately, if you take your hand off the joystick to press the Spacebar, you are likely to get killed.

I have found the easiest way to press the Spacebar while maintaining a steady control of the joystick is to use my foot. I lean back in my chair and set my foot up on the computer desk, with my toes poised above the Spacebar. When I need to throw a grenade, I just press down with my toe. Needless to say, I retain absolute control of my joystick.

David Rees
Bucyrus, Ohio



Bypassing Dongles and Protection Keys

Some commercial programs come with a little plug that must be inserted in a joystick or other port before the program will load. Their makers call these gadgets “dongles,” “keys” or something of the sort; their users have names of their own, many of them not so polite. I’ve learned that I can load Access’s games such as Leader Board and Tenth Frame without using the protection key.

I just plug my Datassette into the cassette port, then proceed to load the game. If I press Play while the game is loading, it skips right through the protection check. You might want to try this on commercial programs of your own.

Michael Kimsal
Mount Clemens, Michigan



Using Key-protected Software on the 128

Upon activating my new 128, I quickly found that I was unable to use the “key” or “dongle” from either Paperclip or The Oracle. (These “keys” are little gadgets that plug into the joystick port; they’re used for program protection. An overhang on the 128’s case keeps them from being plugged in properly). I solved the problem by plugging the key into my joystick extension cable, then plugging the cable into the joystick port. Such cables are available at Radio Shack and at many computer stores.

Arthur K. Yellin
Olney, Maryland



Track and Sector Editors Explained

The commercial and public domain programs known as *track and sector editors* are powerful tools for intermediate and advanced

computerists. They let the user directly read the information written on a disk, and usually allow him to change it. The one I use is called Disk Doctor.

You must be very careful when using these programs, because they have the power to corrupt your disks. Knowledgeable people move very carefully with their disk editing programs; they've been burned a few times in the past. Here are a few points of caution:

- Before using any track and sector editor, read its manual carefully. These are powerful programs that you must understand in depth.
- Learn the program by experimenting on a disk of little value, such as a scratch disk or an outdated backup.
- Always remember that your program has the power to ruin your disk.
- Never, ever use an editor on the original of any commercial program. If you ruin your backup, you can always make another copy; if you ruin your original, you'll have to pay for a new one.

Related Tips in Other Chapters

Chapters 34 and 35 give tips for working with specific commercial software programs. Chapter 36 gives tips for working with word processors.

These tips may also help you in your work with commercial software:

Chapter 15	Hang-up Recovery
Chapter 16	Disconnection Discombobulation
Chapter 30	Game Map Maker

Commercial Game Programs

*“The game is done! I’ve won, I’ve won!”
Quoth she and whistles thrice.*

Samuel Taylor Coleridge
The Ancient Mariner
1798

Every month I receive hundreds of new tips from Commodore game-lovers, making me one of the very few Americans who gets more First Class mail than Third Class. In fact, I get so many game tips that *Commodore Magazine* has started a brand-new column: “Lou Sander’s Gold Mine”. Hundreds of tips from the “Gold Mine” will be printed in a forthcoming TAB book.

In the meantime, here’s an alphabetized collection of the game tips published in “Tips & Tricks” through December, 1988. Don’t get upset if you find one that doesn’t seem to work; many depend on skill. Also, due to the number of programs involved, it’s been impossible to test every tip.



Acrojet

Full flaps (40 degrees) will give you a 100 mph increase in top speed. When flying inverted at low altitude, you will crash unless you retract the flaps. Also, if you turn around and take off heading south, then turn back north, you'll be moving at over 300 mph when you cross the start line. Use full flaps for this "short field" takeoff and get the gear up as soon as you are airborne.

One more quirk: take off heading north and fly toward the largest mountain. After several minutes you'll be in for a very wild ride!

*Kevin J. Fitz-Gerald
Millersville, Pennsylvania*



Aztec

I'm a fanatic of this game, and have gotten the idol 17 times, receiving a reward of \$19,392,550. How? I discovered the following formula for getting to the idol without going through the typical dinosaurs, snakes, spiders, leopards, scorpions, indians and carnivorous plants. You still need to be agile, clever and lucky, but your enemies will be a lot fewer.

When you're in the entrance, press the Spacebar and turn left before going down the stairs. Crawl and place some dynamite, but not too close to the wall. Cover yourself immediately, but don't go down the stairs. When the dynamite explodes, walk toward the hole in the wall and don't stop. Finally, you'll be in the idol's room. Open all the boxes you see, because one of them holds the idol.

If you're trapped in a room and the walls are closing in, just press the F1 key for instructions. To exit, press R. Automatically, the closed wall will open. If you're close to it, you're out.

*Alain Portmann
Santiago de Chile, Chile*



The Bard's Tale II Tip 1

We beat this game when we found the bad guy was really the Sage/Mage. The levels of our characters were: Byron 41 BA, Lord Myron 27 PA, Great Bap 53 AM, Doctor 53 AM, Doc Rock 53 AM, and Sly 41 HU.

*James W. Lee
Seoul, South Korea*



The Bard's Tale II Tip 2

The code for the Dream Spell is "ZZGO". The spell will transport your party to the entry square of one of the seven dungeons in the kingdom, bypassing any passwords or keys you may have otherwise required. Type in the appropriate number to enter these dungeons:

- 1—The Tombs
- 2—Fanskar's Fortress
- 3—Dargoth's Tower
- 4—Maze of Dread
- 5—Oscon's Fortress
- 6—The Grey Crypt
- 7—The Destiny Stone

*Marie McElroy
Santa Ana, California*



The Bard's Tale II Tip 3

When in Garth's Equipment Shop, remove the Bard's Tale II character disk and insert a Bard's Tale I character disk. Then buy an item, and note the very special prices. You can even buy all the segments of the Destiny Wand for \$50,000 a segment. Warning: If you go backward when looking at the list, a jumble of wierd characters will appear.

*Stuart Webb
Yucaipa, California*



The Bard's Tale II Tip 4

This lets you make copies of anything your characters posess, including, and especially, gold pieces. First, get your party into the Adventurer's Guild. Give any character the item you want to copy. If you're copying gold, then pool all the gold to this character. Remove this character from the party, then add the character right back in again.

Now give the item to a different character, and repeat the removing and adding process.

Do this with all six characters, then turn the computer off. DO NOT choose "Leave Game" before doing so.

From now on, when you load the game and add your party, each of your six characters will have the amount of gold you started with.

*Keith Cottrell
Grafton, Ohio*

△ **The Bard's Tale II Tip 5**

If you have an item that has a limited number of uses and you've almost used it up, Sell it back at Garth's. If you repurchase it, the item will have a full allocation of uses.

*Larry Hardy
APO, New York*

△ **The Bard's Tale II Tip 6**

Don't you wish that the "Cat Eyes" spell would point out secret doors? Don't you wish that the "Greater Revelation" spell was indefinite? Then mix the spells!

To get an indefinite lamp that points out secret doors, cast a "Greater Revelation" spell, then cast a "Cat Eyes" spell on top of it. The secret doors should remain visible, and your torch will not go out.

*Steve Harney
El Paso, Texas*

△ **Beach Head**

For monster scores, save your firepower for when the reconnaissance plane gets to the middle of the screen. Keep your finger on the fire button and pump shells into the ball of flame. You'll score 2000 points a hit until the fireball disappears. On the third screen the difference between the angles of elevation is constant. After sinking the carrier the angles left and right are + 16, + 8, + 22, - 28.

*Steve Rohatynsky
Winnipeg, Manitoba
Canada*

△ **Booty**

Hold down keys KEVIN and you can see all of the game screens.

*Edward Teach
Okracoke Inlet, North Carolina*



Bop 'n Wrestle Tip 1

Run to the far right corner of the ring and turn to face your opponent. As soon as he approaches you, kick him in the stomach until his energy is depleted. Then grab him and perform the awful pile driver. He should be dead now, so jump on him for the count of three. Lord Toff, the champion, might need a little more punishment than the less exalted wrestlers.

*P. N. Geek
Madison Square Garden, New York*



Bop 'n Wrestle Tip 2

When taking out your opponent in the one-player version of this great wrestling game, wait until he's facing you, then give him a fast body slam. Go for a quick elbow drop, then wait for him to recover. Repeat the body slam and pin him for the win. Who says computers are for pencil-necked geeks?

*Alex Quevedo
Address Unknown*



Bruce Lee

Make sure you get all the lamps, because it helps in the end, and you can get some pretty high scores. When you're trapped in the corner and the Sumo is charging you, drop straight to the ground and he can't hurt you.

*Steve Rohatynsky
Winnipeg, Manitoba
Canada*



Burnin' Rubber

When you play this game, you have to scratch as many cars as you can; then you get 500 points for each car you scratched. One day I decided not to scratch any cars at all, and I got 50,000 extra points! Because you get an extra life when you reach 100,000 points, my discovery lets you get bonus lives very easily.

It's difficult not to touch any car, but if you practice it becomes easier.

*Arnaud Wiesmann
Jongny, Switzerland*



Championship Lode Runner

This is for those frustrated players who can't get past level 2. Using a track and sector editor, examine track 12 sector 11. You'll see the eight-byte names of the levels, followed by seven other bytes. These seven bytes are the level printed on the screen, the level to be loaded, the number of men, and four bytes for the score.

You can use your editor to change the level printed to the screen and the level to be loaded (screen level number minus 1). For a bigger thrill, change the number of men. FF hex (255 decimal) is the maximum you are allowed.

As always when using a track and sector editor, a mistake can mean disaster. Be sure you know what you're doing before you edit anything on a valuable disk.

*Phil Moores
Chester, Connecticut*



Elite

If pirate ships attack you and fire a missile, respond by targeting a missile on one of them. All pirate ships have ECM systems, so one of the group will ECM your missile and their own at the same time.

*Jean Laffite
Barataria, Louisiana*



F-15 Strike Eagle

When flying an especially long mission, it's easy to run out of fuel within a few minutes of returning to base. One time when returning from a night mission over Hanoi, I started to run out of fuel four map grids away from base. Out of frustration, I started tapping rapidly on the 9 key (100% thrust). I ran out of fuel, but maintained thrust and was able to climb all the way to max altitude, from which I glided to a successful landing.

Later on I learned that while using this trick, you cannot arm or fire weapons or change radar range, although you can still manoeuvre.

*Robert J. Oxley
Marine Corps Air Station
El Toro, California*



Goonies

To get past Screen One, push the chair under the ladder and stand on it to reach the ladder. Then have one Goonie operate the printing press while the other knocks over the water cooler and goes down through the trapdoor. Then rush the other Goonie down to the bottom as quickly as possible.

On Screen Two, you can get past the huge stones by pressing yourself against them when they are all the way down. Keep pressing against the stone while it moves up. When it gets high enough, you'll be able to go underneath it with plenty of time before it falls.

On Screen Three, always know where the sniper is if one of your Goonies is out in the open. A Goonie can avoid the sniper by hiding next to a wall on the third or bottom tier.

To get past Screen Four, have one Goonie hang from the first chain and the other from the third chain. When the cannonball falls through the first hole, rush the upper Goonie down to the second tier and jump up to grab that chain. At the same time, move the first Goonie into position at the right base of the mound in the center of the bottom tier. When your Goonie is in position, let the ball drop through the middle hole. Then place your Goonies into the right position and repeat this process two more times. Be sure to watch the hammer at the lower right-hand corner of the screen. If it's about to strike the bell, place your Goonies in the upper right-hand corner or on the second tier at the point all the way to the left. These are the safe places to hide from bats.

*Patrick McNair
Address Unknown*



Gridrunner II

On Attack of the Mutant Camels, you can jump to the next level by pressing the Control key, the Commodore key and the fire button at the same time. You also get the mystery bonus.

*Mark Horchler
Ellsworth, Maine*



H.E.R.O.

In areas where you might have to retreat, using your laser beam to blast through walls is much safer than using dynamite. You should also slow down before descending. To avoid breaking lights, keep to the left of wide downward passages.

Patrick McNair
Address Unknown

△ ICUPS

If you have a Reset button, you can load ICUPS and press it. Then make these pokes in direct mode:

POKE 3214,234
POKE 3215,169
POKE 3216,0

They will stop alien fire from killing you on the scrolling game.

These will keep your energy from depleting when you touch the aliens:

POKE 45826,234
POKE 45827,169
POKE 45828,0

To restart the game, type SYS 2080 and press Return.

David Metzke
Shepparton, Victoria
Australia

△ Leader Board

This golfing program allows you to make backup copies until they're coming out your ears, but if you misplace the little doo dingy that fits in the back of the machine, your days on the sunny fairways are over!

But don't despair: before you turn on your computer, connect your Datassette recorder, making sure there is no cassette in the machine. Press Play before loading Leader Board, and you'll be back on the links as usual.

Colin J. Adams
Winnipeg, Manitoba
Canada

△ Miner 2049'er

You can change levels very easily if you keep the fire button depressed for about seven seconds at the beginning of the screen.

Hernan Popper
Buenos Aires, Argentina



One-On-One

When playing this popular game against the computer, it is always better to stay near the backboard facing in the direction of your opponent. Most of the time he will run into you, drawing an offensive charging foul. When these fouls accumulate to five, you get to shoot two free throws. The strategy only works when your opponent is the computer, has rebounded the ball and is trying to score on you. Don't try it against a human opponent!

*Stephan Lutz
San Pedro Sula, Honduras*



Pitfall

After loading the game, but before running it, POKE 5383,255 will give you an infinite number of lives.

*Vahan Der Ghazarian
Limassol, Cyprus*



Popeye

If you're having trouble getting off the board in which you catch the hearts, here's your answer: When the screen starts to go, punch the spinach when it is on the second layer, then chase Bluto to the right. When you punch him, he will fly around the room, and you must go quickly to the punching bag. When Bluto is directly beneath the bucket, punch the bag. Bluto will be stuck for the rest of the board, and will be unable to hurt you in any way.

The trick works on any level.

*Robbie Stanford
Corinth, Mississippi*



Quest for Tires

Here's how you can get infinite lives. After loading the program, but before running it:

POKE 7341,99 : POKE 11485,125 : POKE 14864,0

*Vahan Der Ghazarian
Limassol, Cyprus*



Raid over Moscow

If it looks like you're about to crash in the hangar, tap the Spacebar to get you out of there. Press it again for another try on getting your fighter out.

*Steve Rohatynsky
Winnipeg, Manitoba
Canada*



Rambo

While heading back home in your helicopter, you can get some extra speed by pressing the Cursor Down and Return keys while pushing up on the joystick.

*David Metzke
Shepparton, Victoria
Australia*



Seven Cities of Gold

Did you ever lose an expedition because you couldn't find your ship? It's enough to make you swear off this game for good. I avoid the problem by always parking my ship next to a major river. Then if I need to get back in a hurry, I just find the river and follow it back to the ship.

*Jeremy Hubble
Belton, Texas*



Space Harrier

Because this is my favorite game, I've done some research and found some POKES that might be fun to use. Here's how:

Load the program, and when the READY prompt appears, enter:

POKE 1011,248 : POKE 1012,252 : RUN

Press the Return key to load the next part. When it has loaded, enter:

POKE 816,167 : POKE 817,2 : POKE 2096,248 :
POKE 2097,252 : SYS 2061

When the computer resets, make any of the following POKES:

POKE 53277,3	Gets rid of all poles
POKE 14212,234	Speeds up the game

POKE 7673,0 to 15
POKE 7797,0 to 15

Changes shoe and jacket color
Changes trouser color

When everything is finished, enter SYS 2128 to start the game.

*Unknown Contributor
Dubai, United Arab Emirates*



Spy vs. Spy

Go to the room with the airport door and booby trap all the doors. The computer spy won't come here until he has collected all the objects. When he finally comes, he gets killed. This allows you to go back into the room, get the briefcase and leave by the airport door.

*Mata Hari
Leuwarden, The Netherlands*



Summer Games

The World Records for this disk are stored on track 15, sector 5, under the name WR. If you have a track and sector editor, and if you can stand a little innocent cheating, you can change the records to whatever you'd like. Never make such changes on your original disk, because one mistake could put you on injured reserve forever. For safety's sake, use your backup copy.

*David Lu
Tucson, Arizona*



Summer Games I Tip 1

In diving, push the joystick to the right before jumping. Halfway to the water, push it down, left, then up as fast as you can. This will give you a higher difficulty factor and automatically improve your score. On forward and backward dives, hold the joystick to the right before jumping and push up to come out of your dive when you reach the judges. On reverse dives, come out of your position just before entering the water.

In gymnastics, you can get the highest possible score if you press the button to jump, then hold your joystick forward to come off the mat sideways. When your gymnast is in the vertical upside-down position over the horse, press the button. This will give you enough speed to do at least two somersaults.

*Patrick McNair
Address Unknown*



Summer Games I Tip 2

In the pole vault event, it is possible to UNDER the bar at six meters and up, with the jump still counting as a good one. Set your grip for low, and with a little practice you'll be vaulting like a real (?) champion.

*Andy Desak
Monessen, Pennsylvania*



Super Huey Tip 1

If you have trouble with fuel consumption in this popular game, just land and shut off power. When you start up again, you will have a full tank of fuel. I know this works in EXP mode.

*Keith Stevenson
Address Unknown*



Super Huey Tip 2

I love this program, but it's been frustrating to wait so long for the engine to warm up. One day I discovered that if I brought the RPM's down to 120, the initial warmup time was reduced by over half.

*Michael Schroeder
Hudson, New York*



Ultima III

This will help you create "super characters". Using your favorite disk sector editor, read track 16 sector 12 of the scenario disk. This should display the names of your characters; if it doesn't, simultaneously press the shift and Commodore keys. Following each character's name is a series of @'s, then a G if he's "good" or a D if he's dead. After the G are some more text characters leading up to the next name. Change these to 9's, and the associated character will turn into a superman.

Be careful when using this trick, or any trick involving a sector editor. Sector editors change the information recorded on the disk itself, and if you make a mistake you could ruin your disk. So either be very careful and very sure of what you're doing, or work on a backup disk that you can afford to clobber.

*Justin Oldham
Shenandoah, Iowa*

△ **Warhawk**

If you have a Reset button, you can get infinite energy by loading the game, pressing the Reset button, then entering the following:

```
POKE 27090,169
POKE 27091,0
POKE 27092,231
SYS 24604
```

Don't forget to press Return after each entry.

*Andy Carlson
Marietta, Ohio*

△ **Winter Games Tip 1**

In figure skating, keep facing left and falling over to get a maximum score of 6.0 when time is up. Tain't pretty, but it works!

*Alex Chaniotis
APO Miami, Florida*

△ **Winter Games Tip 2**

Here's a trick for everyone who has become frustrated with the one minute figure skating event in this popular Epyx game. Complete one or two jumps, then repeatedly have the skater fall flat on her face. If the judges fall for it as much as your skater did, she'll end up with a perfect score.

*Glen Conway
Lumby, British Columbia
Canada*

△ **Zaxxon**

Type RED at the title screen, and you'll be immune to destruction. You may have to try it a few times until it "takes." When you've succeeded, you'll see the word RED, colored red, just above the ZAXXON logo and just below your previous score. Once that happens, just start and play the game in the usual manner.

Chapter 34

For an interesting game, go to the lowest altitude and fire at the wall. Your missiles will explode, but you'll sail right through the wall, able to blast anything you want.

Dan Clarke
Rochester, New Hampshire

Related Tips in Other Chapters

Chapter 33, "Commercial Software-General Tips" and its related tips contain many valuable items for users of commercial game programs. Many of this chapter's tips mentioned techniques explained in these two items:

Chapter 33	Track and Sector Editors Explained POKEs and Commercial Software
-------------------	---

Other Commercial Software

*“Grace is given of God,
but knowledge is bought in the market.”*

Arthur Hugh Clough
The Bothie of Tober-na-Vuolich
1848

Knowledge bought in the software market can help you in many areas: accounting, drawing, hacking and word processing, to name a meager few.

This chapter has tips for working with specific commercial programs in every area but games. Due to the number of programs involved, it's been impossible to test every tip.



Blitz!

All programs compiled by this compiler start at memory location 2076. The text strings from the original BASIC remain intact and can be examined with a Machine Language Monitor.

*Geoff Buesing
Gahanna, Ohio*



Capture

If you have the Epyx Fastload cartridge enabled, disks made with the Capture cartridge won't load. But if you put this boot program on those Captured disks, it will disable Fastload then load the proper programs:

```
10 REM BOOT
20 POKE 816,165 : POKE 817,244
30 LOAD "BT.*",8,1
```

*John Elliott
Pacifica, California*



Doodle Tip 1

You've probably noticed that this popular program doesn't print images on the printer in the same proportions as it displays them on the screen. You can often make a better printed image by rotating it a quarter turn or expanding it once vertically before printing it. This counters the effect normally produced: a printout that is compressed vertically.

*Jimmy Bednar
Houston, Texas*



Doodle Tip 2

If you have several colored ribbons for your printer, here's a way to get more colorful pictures from your Doodle program. First print the picture in one color, say black, in the normal way. Then change ribbons and use the "photo negative" feature of the Doodle program. Align your paper to its position at the start of the first printout, then reprint it in the new color, which will fill in all the blank spaces.

If your paper is in a slightly different position on the second pass, the resulting color-free and overlap areas can add highlights and texture to your picture.

Steve Martin
Camarillo, California



64 Doodle Picture Loader

Have you ever wanted to load a Doodle picture without loading the main program? Well you can do it with the program in Fig. 35-1.

When you run it, the Doodle picture "FILENAME" should appear on the screen. It's necessary in line 40 to prefix "FILENAME" with "DD" because Doodle assigns those characters as a prefix to the names of all its picture files.

When you use the program, of course, you must substitute your own picture's filename for the "FILENAME" in the example.

Chris Price
Pensacola, Florida

```
10 REM 64 DOODLE PICTURE LOADER - PRICE
20 REM LOADS AND DISPLAYS DOODLE
30 REM PICTURE SAVED AS DDFILENAME.
40 IF A=0 THEN A=1:LOAD"DDFILENAME",8,1
50 POKE 56578,3:POKE 56576,6
60 POKE 53272,PEEK(53272) OR 8
70 POKE 53272,(PEEK(53272) AND 15) OR 112
80 POKE 53265,PEEK(53265) OR 32
```

Fig. 35-1.



Easy Script Tip 1

When I installed a high-capacity SFD1001 as drive 9, I looked forward to using it for my Easy Script files. The manual doesn't say how to use Easy Script with any drive but #8, but believe it or not, other drives are easily accommodated.

When Easy Script first loads, just type a 9 instead of a D at the "Tape or Disk?" prompt. All further operations will default to drive 9. To return to drive 8, simply hit Stop and Restore, then type D instead of 9 at the prompt.

Steve Fishwick
Hamilton, Ohio



Easy Script Tip 2

This very nice word processor is notorious for its slow and head-banging loading procedure. A Fastload cartridge cuts only a few seconds off the load time, bringing it down to about a minute. If you really want to speed up the loading, get your program ISEPIC-ed. The ISEPIC utility isn't 100 percent effective and sometimes takes several tries to produce a working copy, but it can get Easy Script to load in about 15 seconds.

*Jim Brannan
Winter Haven, Florida*



Easy Script Tip 3

The manual for Easy Script does not give instructions for underlining on Commodore's MPS1000 printer. The sequential file 1515 UNDERLINE shows how to do it on the 1515 printer, and fortunately the same commands work perfectly with the MPS1000.

*Clair Black
Nabire, Indonesia*



Easy Script Tip 4

The secret to working with the MPS1000 is to use escape codes to control the printer's built-in capabilities. In Easy Script, escape codes are created by pressing F1 then the Up Arrow character then one or more characters. The characters are case sensitive, so be sure to shift them when called for. Here are the escape sequences to get into and out of several modes:

Emphasized	ESC E and ESC F
Underlined	ESC -1 and ESC -0
Double Strike	ESC G and ESC H
Enlarged	ESC W1 and ESC W0
12 C.P.I.	ESC : and ESC @
Near Letter Quality	ESC x1 and ESC x0
Superscript	ESC S0 and ESC T
Subscript	ESC S1 and ESC T

Unknown Contributor

△ **Fast Hackem**

This program is not only for hacking, but also for speeding up loading. To speed up one of your programs, just boot Fast Hackem and select the Parameters option. After the option is loaded, press the Stop key, enter NEW, and clear the screen. You'll then have a super fast loading program for the 1541.

With this trick, I've been able to load Pac Man in under ten seconds, and an entire high-resolution screen in about eight.

*John Stuart
Quezon City, Philippines*

△ **Fastload Cartridge Tip 1**

It's only partially explained in the instructions, but you can use Fastload to automatically load and run the first program on a disk's directory, no matter what it is. If you have many programs on a disk, make the first one a menu program and you'll never have to type LOAD and RUN again.

An easy way to load another BASIC programs, without using a menu, is to type this, where [RUN] is the shifted Run/Stop Key:

`/PROGRAMNAME* [RUN]`

You have to type the asterisk, even if you've typed the program name in full. Of course, you can type only enough of the program name to make it unique, and the asterisk will act as a wild card.

*Jack M. Hirsch
Rockville, Maryland*

△ **Fastload Cartridge Tip 2**

Drive Not Ready errors usually result from your disk door being open or not having a disk inserted. But sometimes the drive just decides not to be ready, and that can be a real aggravation. If you get one of these errors while using the Fastload cartridge, you can usually correct it by entering @U; or @U+. The same commands often work when your drive is recalcitrant in other ways.

*Rickey Joe
Palisades Park, New Jersey*

△ **Fastload Cartridge**

If you wish to reset your computer but keep Fastload intact, enter the monitor by entering an exclamation point, then enter G FCE2 at the period prompt.

*Phil Brubaker
Russell, New York*

△ **Fleet System 2 Tip 1**

An undocumented feature of this program is the high-speed cursor you can get if you press F1 prior to using the Cursor Up or Cursor Down keys.

*Scott A. Warner
Bethel Park, Pennsylvania*

△ **Fleet System 2 Tip 2**

This excellent word processor lacks a command to take you to the end of the document, but you can fake one if you know what to do. Just use the <FCN> h command, without using <FCN> f first. In effect, you're asking the computer to hunt for "nothing". It immediately searches the document, finds nothing, and stops at the end of the page.

I find this much easier than scrolling up and down or using the jump feature, especially when I'm opening up a document in progress and want to continue where I left off.

*Harry Calhoun
Pittsburgh, Pennsylvania*

△ **Fontmaster**

This applies only to the original version of this program; newer versions have eliminated the need for it. The program lets you create or modify typefaces and save them to disk, and although XETEC discourages it, these new typefaces can be saved onto the original Fontmaster disk, where there is plenty of room for them.

I discovered that the original fonts cannot be changed, replaced or erased, but that their names can be changed. So I used the Fontmaster commands to rename the ENGLISH font as ENGLISH0, then saved my modified font as ENGLISH. The result is that the [CMDR F3] command, which automatically loads eight of the manufacturer's fonts, now loads

my ENGLISH instead of their ENGLISH0. When I want to use the original version, I just load it individually using its new name.

By the way, the Fontmaster disk may have a few typefaces not mentioned in the manual. Use the F8 key to find their names so they can be loaded individually.

*Bernard J. Sussman
Washington, D.C.*

○△ **General Ledger**

Info Designs has created a useful and low-cost accounting package here, but it has a bug when used with my Gemini 10X printer and Card? + G interface. Each report begins with a line of asterisks; but with my printer setup, the asterisks are in an expanded typeface that won't fit on a single line.

My solution is to turn the printer off and back on again right before printing a report.

I also found that by adding a line to the boot program, I could put my interface into transparent mode by software. (Previously, I had to set my dipswitches each time I used a different program). The line I added was:

```
12 OPEN4,4,24 : PRINT#4 : CLOSE4
```

*Jim Berg
Greenville, South Carolina*

△ **GEOPAINT**

Because this drawing program's screen output is longer vertically than its printer output, drawings become squashed on the printout.

One way to fix this is by adjusting the vertical height on your monitor until the screen picture matches the printout. But what if, like myself, you use a TV set for a monitor?

A solution can be found in the X and Y measures at the bottom of the screen. When they are equal, you have a square. Draw one and use it as a model. Because it won't look square on the screen, try to squash everything else to match it. Even if you don't squash perfectly, your printed output will be closer to what you see on the screen.

*Juan Rodriguez
Abington, Pennsylvania*



GEOPUBLISH

It appears that GEOPUBLISH addresses files not only by name, but also by position. Make sure that all the text, photo scrap and font files you plan to use are on the disk that you're using.

I deleted a file because I was running out of disk space, and all of a sudden I got errors all over the place whenever I tried to ripple or edit any text.

*Fortytude
Quantum Link*



GEOS Tip 1

If you have finished detailed work in the pixel edit mode, but there is more to be done outside the magnified area, just click on the four arrows in the Geopaint tool box. This will take you out of pixel edit mode and let you move the editing box on the screen to the place you want to continue your editing. This is easier and faster than double clicking the pencil icon on and off to get into and out of pixel edit mode.

*Chris McAlister
Knoxville, Tennessee*



GEOS Tip 2

GEOS allows customized preferences to be put onto each work disk. But having to create one from scratch for each disk is tiresome, and copying with the icons is clumsy. An easier and faster way is to load up the Preference Manager and load the preferences you want to copy. Remove the original disk, then insert the destination disk, then save the preferences onto that disk. You can do this for each disk you have. Before you exit, place the original disk back into the disk drive.

*Leon S. Brandy
Christiansted, U.S. Virgin Islands*



GEOS Tip 3

On booting GEOS, the default print driver is the first one on the disk. You can make your own driver the default by doing this: click the first print driver's icon once, then go to the command bar and under the FILE select the Rename option. Then simply rename the file to that of your own printer's print driver. Because GEOS searches only for the name, it will now select your printer as the default.

If you have a single disk drive and a memory expansion module, you have the fastest disk copying system you've ever seen. Copy your source disk (from the desktop under Disk) into your expansion module. Then insert your destination disk and copy the expansion module into the physical disk drive.

*John Kuhne
Newmarket, Ontario
Canada*

△ **GEOS Tip 4**

When using GEOS, the printed output is slightly shorter than what is shown on the screen. As a consequence, drawings become squashed and out of proportion. To fix this, just adjust the vertical height on your monitor until the screen is just as squashed as your printed picture. From then on, anything that you draw on the screen will be duplicated identically on the printer.

*James Williamson
Marietta, Georgia*

△ **GEOS Tip 5**

Not mentioned in the GEOS manual is the existence of several characters not normally available from the 64 keyboard. These characters are part of the standard ASCII character set, but not part of the Commodore set. You'll get them by pressing the indicated keys with the Commodore key held down.

Left curly bracket [CMDR :]
Right curly bracket [CMDR ;]
Underline [CMDR -]
Backslash [CMDR /]
Vertical line [CMDR UP ARROW]
Tilde [CMDR *]
Left apostrophe [CMDR @]

*Francis Volpe
Pittsburgh, Pennsylvania*

△ **GEOS Tip 6**

This popular program works very slowly with only one disk drive. Adding a second drive gives a big improvement in speed. Adding a RAM

expander also speeds things up; my new 1764 was a very worthwhile investment.

*Lucy S. Terrier
Alton, Illinois*



GEOS Tip 7

If you use an 80 dpi dot matrix printer to print out GEOWRITE files, you know that what you see on the screen is exactly what you get on the paper. But if you use an Apple LaserWriter or use the services of the GEOS Laser Printing Center, your printed output does not always look as you expect it to.

This is because even the GEOS laser fonts cannot perfectly duplicate on the screen the LaserWriter's letter proportions and spacing. The differences are most pronounced when centering long lines of outline text — the LaserWriter lines will be printed too close to the left margin and definitely not centered.

To center outline text for printing on the LaserWriter, first center the text and change to normal style (bold for bold-outline). Then place a tab above the beginning of the line of text. Next, left justify the text and change to outline (or bold-outline) style. Finally, move the text over to the tab with [CTRL I]. If the text wraps around to the next line, you'll either have to move the right margin rightward or shorten the line of text.

I've found this information useful when using GeoPublish to edit our local Commodore users group newsletter.

*Dave Peterson
Ames, Iowa*



GEOS Tip 8

To make the Panasonic KX-P1080i printer work with GEOS, go to printer setup and change the printer driver to MPS1000.

*Saeed Sadeghi
Granada Hills, California*



GEOS Tip 9

Though GEOS can be used to make very nice signs, notebook covers, and so forth, it lacks the capability of making borders. Print Shop and Printmaster, of course, can create a wonderful selection of borders.

To create a GEOS document with a border, first use Print Shop or Printmaster to print a border on a sheet of paper. Then roll the paper back into your printer and use GEOS to print the text.

Peter Lin
Irvine, California

☆ **GEOS 128**

This program includes many improvements that are not available in the 64 version. One that you can use very frequently involves the all too familiar OK prompt.

The OK prompt appears when GEOS 128 asks you to confirm an error such as Missing Disk, as well as when it asks you if you are sure you want it to execute a certain command.

When you get the OK prompt, you do not have to drag the mouse or joystick over to the prompt to signify "OK". Just hit the Return key instead.

Brendan Schulman
Winnipeg, Manitoba
Canada

△ **GEOWRITE**

To get rid of a troublesome tab on the ruler, just touch it once with the pointer to highlight it, then drag it beyond the margin indicated by the M symbol.

To avoid formatting problems in GEOWRITE V2.0 + , always insert your headers and footers before you begin. Otherwise GEOWRITE will occasionally botch headers inserted later, and when you try to reformat your document the system will crash.

John Kuhne
Newmarket, Ontario
Canada

△ **Hesmon**

The normal command to enter Hesmon is **SYS 64738**. Unfortunately, this command also disables the Fastload cartridge, if you are using one. To enter Hesmon without causing this problem, use the command **SYS 36466**.

Dan Kollmorgen
O'Fallon, Missouri



Koala Painter

You've probably noticed the special symbol on the front of all the Koala Painter filenames. It represents the color for orange, and can be produced by holding down the Commodore key and pressing the 1 key. (You have to be between two quotation marks, of course!) So if you need to access these files outside of Koala Painter, just make this character the first one in the filename.

*Jim Partin
Cincinnati, Ohio*



Magic Desk

If you don't want to use the joystick in Port 2 to move the finger, you can use these keys to move it:

[CTRL A] Down
[CTRL D] Left
[CTRL LEFT] Down
[CTRL G] Right
[CTRL J] Fire Button

*James Schnur
Redington Beach, Florida*



MicroIllustrator

In this Tech Sketch program, the magnify mode is very slow to move until the third or fourth time you hit the Shift key. Hitting and releasing Shift Lock has the same result and never fails.

*ZaphoddB
Quantum Link*



Multiplan

I have used the Epyx version of Microsoft Multiplan for a long time with my 64, and have found it to be one of the most useful programs I own.

When I recently got a 128, I was eager to see how Multiplan looked on its 80-column screen. It was wonderful to see twice as many columns as before, but to my pleasant surprise this wasn't the best improvement with the 128.

Multiplan runs much faster on the 128 than it does on the 64, and the difference is quite noticeable, even when working with fairly small spreadsheets. Not only that, but switching to 128 mode eliminates all those accesses to the disk drive. In fact, with the 128 you don't even need to transfer the MP.SWAP file to your data disks!

*Kathleen Mead, CPA
Westerville, Ohio*



The Newsroom

Borders are a common feature of desktop publishing packages, but the popular Newsroom program seems to lack this option. But you can create a very nice border by selecting the Old English font and typing a series of dashes! Alternate other punctuation marks for an even more decorative look.

Borders can also be created with the BOX function. Frame your work with a box then frame it again with a second smaller or larger box. You can fill the area between them, if you'd like.

*Craig Edward Given
Chattanooga, Tennessee*



Paperclip Tip 1

This powerful and popular word processor contains a bug that can cause a crash in the middle of editing. When the end of the file happens to fall in the 40th column, any deletion that involves the final character will cause the program to lock up. The only way out is to turn the computer off, losing everything in memory.

If you keep a Return character as the last character in your document, this bug will never cause a crash.

*Art Kohn
Duke University
Durham, North Carolina*



Paperclip Tip 2

Paperclip 64/128's dongle does not plug into the intended Port 1 of the 128, due to the sculptured side of the machine. Paperclip II corrected the problem by having its dongle plug into Port 2. Unfortunately, this alienates all Paperclip64 processed files!

Even in Port 2, the dongle fits poorly and sometimes drops out because it protrudes like a sore thumb from the side of the 128.

The solution for all dongles is to make a joystick cord extension, or more simply, to buy a joystick extension cord (Radio Shack #270-1705, available for \$4.99 in the 1988 catalog). Plug it into the appropriate port, route the extension along the existing power supply cord, and let the dongle dangle out of sight.

*Edmund G daSilva
Rio de Janeiro, Brazil*



S'More Cartridge

If you use this popular 64 cartridge with your 128, you may be frustrated because the computer doesn't automatically enter 64 mode when it's turned on. And if you enter 64 mode by using the GO 64 command, the cartridge still won't start up properly.

The solution is to enter 64 mode by holding down the Commodore key at power up, or by holding it down and pressing the Reset button.

*David Schreiber
San Jose, California*

Related Tips in Other Chapters

These tips can help you with commercial software programs:

Chapter 8	Colorful Printouts Home-made Envelopes Greeting Card Envelopes
Chapter 21	LOGO and Print Shop
Chapter 22	Kaypro CP/M Programs on the 128
Chapter 33	Track and Sector Editors Explained

Chapter 33 also contains other general tips on working with commercial software. Chapter 36 has many tips on working with word processing programs.

Word Processing

“Good words are worth much, and cost little.”

George Herbert
Jacula Prudentum
1651

When they were prepared for *Commodore* magazine, the good words in this book were processed on my \$750 Commodore system, using the \$29.95 Easy Script word processing program. When I processed them for this book, I moved up to a \$3,750 IBM-compatible and the \$229 Microsoft Word program.

The MS-DOS system cost five times as much as the Commodore. For tasks short of writing a book, it is certainly not five times as useful.

Whichever word processor you prefer, the tips in this chapter can help you use it more effectively.



Word Processing Placeholder

If you're working on a long document and need to quit for the moment, it's sometimes hard to remember where you were working when you quit. The same trouble arises when you need to move temporarily to the top or bottom of the document.

You can make it easy to find your place by typing one or more special characters before you move away. To return, just use your software's "find" function to search for the special characters.

I often use two asterisks, dollar signs or other keyboard characters as my placeholder. Anything at all will work, provided it isn't repeated elsewhere in your document.

*Arleigh
Quantum Link*



Convenient Word Processor Formatting

Sometimes when you are going to be using the same text formatting commands on several different documents, it's easier to make a file out of them. Then, whenever you want a document using that format, you can load the file and append your document text after the formatting directives.

*Matthew Ettus
Rego Park, New York*



Unslashing Your Zeroes

These days, many printers are capable of near-letter-quality or better in their output, and it's common to use them for printing important documents. But sometimes, even though the print looks marvelous, the zeroes have that tell-tale slash through them. This touch of nerdhood is a no-no in many contexts, especially in a resume.

The way out? Use your global search-and-replace feature to change every zero to a capital "O". Goodbye, slashes.

*Stephen Knapick
New Rochelle, New York*



Video Proofreading

It's hard to do a good proofreading job on my 40-column word processing screen, because the words are wrapped around the right-hand margin.

So when I'm ready to proofread my document, I temporarily set the margins to 1 and 39, then print it out to my video screen. The software eliminates the word splits, making it much easier to spot errors and places for improvement. When I'm ready to print on paper, it's easy to reset the margins to something more suitable for an 80-column output.

Karen E. Crum
Waverly, Kansas

○△☆ **Improving Justification**

Like many word processors, my GEOWRITE 2.0 offers both word wrap and right and left justification. But when a long word on one line is wrapped to the beginning of the next line, large and unsightly spaces are created when the first of these is justified. The adjustment of these deficiencies was annoying and time consuming until I discovered a simple procedure to eliminate the problem.

As I type and approach the end of the line, I note any long word that would normally be word-wrapped. I type the word through its initial syllables until I reach a suitable hyphenation point near the end of the line. Then I type a hyphen and a space, followed by the rest of the word. The space triggers the wrap function and the word is appropriately divided and hyphenated, and the justification looks much better than it did the other way.

Initially the procedure may seem awkward, but it soon becomes automatic, and the results are well worth the effort.

Charles W. Seager
Address Unknown

○△☆ **Automatic Paragraph Numbering**

When creating a document on your word processor, it's often desirable to number paragraphs or other sections of your work. But when you use the word processor to move a section from one place to another, the numbering gets out of sequence and you have to correct it manually. Not only is this a nuisance to do, but it's an invitation for errors to creep into your otherwise perfect work.

But it's easy to get your word processor to do the numbering for you, and of course it never makes mistakes! The secret is to pretend that your document is a form letter, and establish variable blocks wherever the numbers are to appear. When outputting the text to the screen or

to the printer, link your file to a variable data file filled with consecutive numbers. (Mine is titled "NUMBERING".)

*Walter R. Beardsley
Santa Maria, California*



Printing Labels

It's often difficult to get labels to print on a word processor or database program, because these applications are oriented to full size sheets of paper. Often, changing the program's sheet size will let it work easily with your label stock. Many labels are one inch high and hold six lines of printing. For these, just set your page size to six lines, and watch how things improve.

In Paperclip, the command to send is PP6:PG5. Other word processors have similar commands.

*Andre Leduc
Hawkesbury, Ontario
Canada*



Organizing Word Processor Files

If you do much writing with your word processor, it's easy to accumulate a long list of files stored on your disk. Pretty soon, it gets hard to remember which file is which, and it gets very hard to retrieve what you want from the disk.

The 1581 drive happily increases disk storage space, but it makes this problem even worse. Here's my solution:

- 1) Format a blank disk.
- 2) Create a blank document and save it with the name DIRECTORY.
- 3) When saving a letter or other document, name it with the date and a short identifying name. For example, 070488JSMITH might be a letter written to John Smith on July 4, 1988.
- 4) As soon as you have saved such a document, load DIRECTORY.
- 5) Add the new letter's name to DIRECTORY, adding a few keywords. In our example, you might add 070488JSMITH - Amiga - Software.
- 6) Resave DIRECTORY.

When it's time to retrieve your letter, load DIRECTORY into your word processor and use the Search or Find feature to look for the keywords you remember.

QTUTOR bar
Quantum Link

Related Tips in Other Chapters

Chapter 33 contains general tips on working with commercial software. Chapter 35 has tips on several commercial word processing programs.

These tips are also of use in word processing:

Chapter 4	Printer Ruler
Chapter 30	64 Easy Word Processor
	128 Easy Word Processor

Index

1902 monitors, improving
performance of, 49
6502 microprocessors, 142

A

abbreviations, 11, 12
 BASIC, 34-35
 zero, 34
accessories, build-it-yourself
 projects, 127
accounting package, 369
Acrojet, 350
address finder utility, 313, 314
ADSR table maker, 244
alcohol, cleaning drives with, 30
alphabet trainer, 308
Amiga computers, 131
amusements and delights, 276-298
antennas, filters for, 47
apostrophe (F7), 41, 174
appender, 169, 188, 339, 340
application programs, 299-310
array subscripts, 154
ASC, 42
ASCII, conversions in, 209
assembler language programs, 138,
 192
assemblers, 162

AUTO fix, 144
auto-centering routine, 226
autoanswer disabler, 102, 103
automatic LOAD-RUN, 42
Aztec, 350

B

backup disks, 92
 DOS Shell programs, 95
BANK command, 144
Bard's Tale II, The, 350-352
bars, screen, 221
bases
 identification of, 250
 universal conversion program
 for, 255
BASIC, 138, 144-161
 abbreviations for, 34-35
 compiled, 192
 disk commands, 147
 keywords for, 34-35
 learning to program with,
 139-141
 linker code, 170
 linker utility, 336
BASIC ROM, memory locations,
 166
basic.basic.sda program, 140-141
batting averages, 308

Beach Head, 352
beep routine, 237
BEGIN, 144
BEGIN...BEND, 144
bell tone, 236, 237
binary key decoder code, 170
binary numbers, 250-253, 256
Bird Art, 295
Blitz!, 364
blocks-free reader utility, 313
books, 3, 7-19, 23
Booty, 352
Bop 'n' Wrestle, 353
border fireworks code, 171, 225
border killer code, 128, 189, 233
bottom-feed printers, 56
brackets, use of, 11
Break-Fast, 293
Bruce Lee, 353
bugle calls sound, 240
bugs (see debugging)
build-it-yourself accessories, 127
bulletin board systems, 109-110
Burnin' Rubber, 353
burning in, new equipment, 128
Burrow, 277
business use, 22
buzzer sounds, 236
bypassing magazine entry

- programs, 13
 - bytes, 203
- C**
- C language, 192, 196
 - Caps Lock key, reading position of, 43
 - Capture, 364
 - cartridge expansion boards, 126
 - cartridges, 126-127
 - cassette tape recorders, 73-75
 - Castle Siege, 284
 - centering text, 215
 - automatic routine, 226
 - Championship Lode Runner, 354
 - CHAR, 146
 - character format conversions, 209
 - character sets, switching between, 36
 - "charge" melody, 239
 - checksum programs, 9, 12, 15
 - apostrophe key (F7) for, 41
 - Chopper Pilot, 288
 - CHR\$ conversions, 42
 - cleaning, 30
 - keyboard, 36
 - printer printhead, 58
 - CMDR key, 11
 - COBOL, 192
 - COLOR command, 160
 - color display, 46
 - changing, 218
 - characters, changing, 218
 - flipper for, 221
 - function keys, 222
 - monochrome monitor for, 216
 - restore colors utility, 342
 - SPREDEF setting and, 160
 - color flipper code, 171
 - color sys code, 171, 223
 - COMAL, 194-195
 - combinations, permutations and, 259
 - commands, BASIC disk, 147
 - Commodore Logo, 296
 - computer rooms, 25-32
 - computer-related income, 22
 - Computerfacts, 119
 - computers, 130-136
 - Amiga, 131
 - bargain prices for, 21
 - dead, 117, 119
 - fuse replacement in, 117
 - generic, 131
 - IBM-Commodore compatibility, 131
 - new equipment burn in, 128
 - PC clone, 132
 - Plus-4, 133
 - schematic diagrams for, 119
 - sharing use of, 29
 - shorting in, 117
 - used vs. new, 21, 23
 - connectors, 111-115
 - DIN, 112
 - expansion port, 114
 - joystick port, 113
 - labeling, 28
 - modem, 101
 - polarization of, 114
 - port, 113
 - CONT statement, 148
 - debugging and, 266
 - endless loops and, 146
 - conventions and shows, 4, 5
 - conversions
 - binary to dec, 253
 - binary to hex, 252
 - character format, 209
 - dec to binary, 253
 - dec to hex, 254
 - Fahrenheit to Celsius, 307
 - function keys for, 42
 - hex to binary, 252
 - hex to dec, 254
 - Roman numerals to Arabic, 255
 - universal base, 255
 - CONVT routine, modification of, 200
 - CP/M operating system, 197-201
 - disk swapping with, 200
 - introduction to, 198
 - Kaypro programs for 128 in, 199
 - printers and, 200
 - reboot, 199
 - screen width changes with, 199
 - software for, 199
 - CTRL key, 11
 - cursor, flashing, GETKEY and, 150
 - CXPRINTE.ASM, modification of, 200
- D**
- DATA statements, 13, 144, 161, 206
 - debugging, 268
 - errors in, 263
 - function key to produce, 41
 - shortening, 146
 - database programs, 139
 - DATAMAKER.DEC utility, 329, 330, 331
 - Datassette, 73-75
 - date and time in filenames, 87
 - dead computer, 117, 119
 - debugging, 262-270
 - DATA statements and, 268
 - deceptive lines for, 264
 - error reported on wrong line in, 265
 - finding bugs for, 263
 - helper for, 266
 - printer hex mode in, 54
 - STOP and CONT for, 266
 - DEC, 42
 - decimal numbers, 250-254
 - Defensive Driving, 287
 - Delete key, right of cursor use of, 40
 - delete right code, 172
 - deleting lines, 212
 - Dice Machine, 301
 - DIN connectors, 112
 - DIP switches, settings for, 52
 - directories
 - basher utility for, 318
 - disk, by filetype, 90
 - doublespaced printout of, 91
 - loading from, 89
 - printing out, 91
 - program purpose and, 142
 - programs in, 88
 - protected, reading of, 92
 - slowing down display of, 33
 - viewer utility for, 312
 - DIRECTORY command, 147
 - disconnection, serial bus, 124, 125
 - disk defaulter code, 172
 - disk error reader code, 173
 - disk operations, 85-99
 - Disk Signer program, 10, 318
 - disks, 203
 - backups for, 92
 - BASIC commands for, 147
 - defaulter utility for, 320
 - double-density, 77
 - DSDD, 78
 - envelopes for, 81, 82
 - error reader utility for, 321
 - filetype directories for, 90
 - floppy, 76-84
 - high-density, 77
 - identification of, 79
 - labeling, 80, 81
 - library tips for, 79-80
 - mailers for, 82
 - memory capacity of, 77
 - notching, 30
 - purchase guide, 77
 - quadruple-density, 77
 - recycling, 21
 - sectors in, 77
 - signer utility for, 10, 318
 - SSSD, 77
 - storage of, 30, 82, 83, 84

Index

- using opposite side of ("flippies"), 78-79
 - write-protection for, 30, 78, 80
 - displays, 45-50
 - 80-column line space changer, 228
 - antenna filters, 47
 - auto-centering, 226
 - border fireworks code, 225
 - border killer, 233
 - centering text in, 215
 - character color changes, 218
 - color and brightness adjustments for, 46
 - color flipper, 221
 - color sys code, 223
 - color, SPREDEF setting and, 160
 - easy windows, 223
 - end writer, 233
 - exciting letter display feature, 216
 - explosion, 232
 - F-key colors, 222
 - Fancy Printing program for, 216
 - glare reduction, 46, 48
 - Hi-Res helper utility for, 341
 - interference on, 46, 47
 - monochrome monitor "color," 216
 - multicolor character, 37
 - musical staff, 246
 - pausing graphics, 230
 - photographing, 49
 - PRINT@ code, 224
 - programming of, 214-234
 - restore colors utility for, 342
 - RGB monitor extension, 49
 - screen bars, 221
 - screen color change, 218
 - screen framer, 220
 - screen marking pens for, 48
 - screen mode swapper, 230
 - screen POKEs, 227
 - screen saver, 215
 - screen scrambler, 219
 - simultaneous text and graphics, 228
 - slowing down (CTRL key), 33
 - special effect, 225
 - sprite rotator, 231
 - symmetrical scaling, 230
 - test pattern utility for, 325
 - TV-set, 46
 - vibrating screen, 215, 219
 - video monitor switch for, 50
 - division tutor, 310
 - DLOAD, 147
 - documentation, commercial software, 345
 - dongles, 347
 - Doodle, 364-365
 - DOS Shell programs, 95
 - DOS Wedge
 - defaulter utility for, 322
 - disabler utility for, 321
 - slowing programs with, 95
 - double-density disks, 77
 - DPS1101 printwheels, 67-68
 - DRAW, shortcut for, 147
 - drives, 203
 - 1571, quirks of, 97
 - cleaning, 30
 - double-sided mode for, 97
 - motor shutoff in, 98
 - reset of, 93
 - speed of, 96
 - storage capacity of, 96
 - transporting of, 94
 - types of, 77
 - unsticking, 93, 94
 - DSAVE, 148
 - DSDD disks, 78
 - dump, 186
 - variable, debugging and, 267
 - variable, utility for, 331
- ### E
- Easy Script, 365
 - easy windows code, 173, 223
 - easy-load filenames, 89
 - editing, precautions for, 268
 - electronics construction projects, 127
 - Elite, 354
 - ELSE, 144
 - ELSE BEGIN, 144
 - END statements, 15, 148
 - end writer, 233
 - endless loops, 123, 146
 - envelopes
 - disk, homemade, 81, 82
 - home-made, 63, 64
 - error checking, 12
 - Error Messages, 297
 - error reader, disks, utility for, 321
 - errors (see also debugging), 263
 - published programs containing, 10-11
 - escape sequences
 - graphics mode, 44
 - printer, 56-57
 - reverse character, quote mode, insert mode, 44
 - European siren sound, 238
 - Evasion, 296
 - execution, speeding up, 206
- ### F
- expansion ports, connectors for, 114
 - exploding screen effect, 232
 - explosion sound, 238
 - exponential notation, 34
 - eye care, 48
- ### F
- F-15 Strike Eagle, 354
 - F7, apostrophe key use of, 41, 147
 - factorials, 258
 - Fahrenheit to Celsius conversion, 307
 - Fancy Printing program, 216
 - fanfare melody, 239
 - fanfold paper, 61
 - Fast Hackem, 367
 - Fastload Cartridge, 367, 368
 - filenames
 - date and time in, 87
 - easy-load, 89
 - TI\$ in, 87, 88
 - files
 - "goofer" utility for, 320
 - renaming utility for, 312
 - word processing organization, 380
 - filetype directories, 90
 - Fireworks, 297
 - Fleet System, 368
 - flipper, color displays, 221
 - "flippy" disks, 78-79
 - Flower Power, 279
 - Fontmaster, 368
 - FOR...NEXT, 149, 155, 207, 212
 - FORTH, 194
 - FORTRAN, 194
 - framer, screen, 220
 - function keys
 - color code for, 222
 - colors machine language code for, 175
 - conversions performed with, 42
 - disabling, 43
 - reassignment-original assignment of, 41
 - fuses, 117
 - Fuzzy Caterpillar, 286
- ### G
- games, 276-298
 - commercial, 349-362
 - map maker for, 300
 - no-hands playing with, 346
 - General Ledger, 369
 - GEOPAINT, 369
 - GEOPUBLISH, 370
 - GEOS, 370-373

- GEOWRITE, 373
 GET statements, 43, 203
 GETKEY, 43, 150
 glare screens, 26, 46, 48, 49
 Goonies, 355
 GOSUB, 9, 268
 GOSUB/RETURN, 212
 GOTO, 9, 148, 150, 206, 268
 Grand Finale Fireworks, 297
 graph maker, sprite, 302
 graph paper, program for, 305
 graphic symbol keys, 9
 graphics
 Hi-Res helper utility for, 341
 pausing, 230
 symmetrical scaling, 230
 text simultaneous with, 228
 graphics mode, quick escape from, 44
 greeting cards, envelopes for, 63, 64
 Gridrunner II, 355
- H**
 H.E.R.O., 355
 hands-on time, 2
 happy birthday melody, 242
 hardware, 121-129
 heat-sensitive paper, 62
 helicopter sound, 237
 HELP key, 43
 Hesmon, 373
 hex mode, printer, 54
 HEX\$, 42
 hexadecimal address, 168
 hexadecimal numbers, 42, 250-254
 Hi-Res helper utility, 341
 Hidden Message, 295
 high-density disks, 77
 highlighters, 29
- I**
 I-O devices, memory locations, 166
 IBM computers, Commodore
 compatibility with, 131, 197
 ICUPS, 356
 IF statements, 150, 207
 IF...THEN...ELSE, 144
 illegal characters
 debugging help with, 266
 INPUT and, 151
 INPUT statements, 37-39
 default value for variable in, 152
 error correction for, 209
 illegal characters and, 151
 question mark and, 151
 unwanted characters in, 153
 insert mode, escape sequence, 44
- INT, 154
 interface
 printers, power supply for, 52
 printers, SX 64 and, 136
 interference, 46, 47
- J**
 Jaws LXIV, 281
 joysticks, 69-72
 connectors for, 128
 inaccessible ports for, 113
 port connectors for, 113
 Jumbles, 282
 jump table, memory locations, 167
 justification, 379
- K**
 kernal jump table, memory locations, 167
 kernel ROM, memory locations, 167
 key-protected software, 347
 keybeeper sound, 176, 243
 keyboard errors, joysticks and, 71
 keyboards, 32-44
 repair of, 135
 sensing special keys for, 207
 keystroke abbreviations, 11, 12
 keywords
 BASIC, 34-35
 changes in 64 and 128, 35
 Koala Painter, 374
- L**
 labeling
 cables, 112
 connector, 28
 disks, 80, 81
 joystick, 70
 program, 10
 labels
 inexpensive, 63
 printer alignment of, 55
 printer gum-up and, 62
 printer program for, 305
 printing, 380
 Lampman, 281
 languages, 192-196
 Leader Board, 356
 learning, 1-7
 lighting, 26
 lightning detector, 28
 line counter, 213
 line deletion, temporary, 204
 line number suppression, 212
 line spacing, 80-column, changing, 228
 linker utility, BASIC, 336
- LIST**
 automatic use of, 204
 bell tone for, 156
 disabling of, 155
 double-spaced printout of, 154
 indentations in, 155
 stopping, 204
 literature, reader's guide to, 3
 LOAD program
 automatic, 42
 easy execution of, 90
 loading
 from directory, 89
 machine language utility for, 338
 slowdown of, 94
 locked files, 96
 lockups, 123
 logarithms, 260
 LOGO, 194, 195
 looping
 CONT statement to create, 146
 termination of, 149
 Lottery Assistant, 300
 Lotus, 139
 lullaby melody, 240
- M**
 machine language, 161-191
 appender code, 128, 169, 188
 BASIC linker, 170
 binary key decoder code, 170
 border fireworks code, 171
 border killer code, 128, 189
 code summary of, 168-191
 color flipper code, 171
 color sys code, 171
 delete right code, 172
 disk defaulter code, 172
 disk error reader code, 173
 easy windows code, 173
 F-key colors, 175
 F7 apostrophe code, 174
 keybeeper code, 176
 learning, 142
 loader for, 176, 338
 memory locations for, 165-168
 memory maps of, 165
 mirror code, 177
 mnemonic for, 168
 monitors for (see machine language monitors)
 operands, 169
 pauser code, 178
 PRINT@ code, 178
 printouts from monitors using, 162
 quick erase code, 179
 REMVERT code, 179

Index

- saving utility for, 315
 - screen bars code, 180
 - screen effect code, 181
 - screen printing, 165
 - screen scrambler code, 191
 - search code, 181
 - SIXPACK PLUS code, 183
 - string printer for, 164
 - timer code, 185
 - variable dump code, 186
 - machine language monitors, 162-163
 - machine type detector, 208
 - magazine entry programs, 12-17
 - magazines, 3, 23, 7-19
 - Magic Desk, 374
 - mailers, disk, 82
 - manuals, 2
 - Map Maker, 300
 - mathematics programming, 248-261
 - memory maps, machine language, 165
 - memory saver utility, 314, 317
 - menus, 88
 - MEP Autoloader, 16, 18
 - MEP128.ML, 16
 - MEP64.ML, 16
 - Microillustrator, 374
 - Microsoft Word, 139
 - Mighty Mite, 279
 - Mind Reader, 277
 - Miner 2049'er, 356
 - Mirror, 177, 288
 - ML saver utility, 315
 - mnemonics, machine language, 168
 - mode guard utility, 341
 - mode swapper, screen, 230
 - modems
 - autoanswer, disabling, 102, 103
 - connections for, 101
 - hang-up and disconnect for, 104
 - silencer for, 103
 - speed dialing, 101
 - telephone cable marking for, 101
 - telephone extension for, 101
 - modifying magazine entry programs, 15-17
 - Modula-2, 194
 - money-saving tips, 20-24
 - monitor shelf, 27
 - monitors
 - machine language (see machine language monitors)
 - monochrome, "color" display on, 216
 - mouses, 69-72
 - MPS802 printers, Blue Chip printer equivalent to, 58
 - MPS803 printers, 57-58
 - MS-DOS, 139, 197, 198
 - multicolor characters, 37
 - Multiplan, 139, 374
 - multiple printer connection, 53-54
 - multitasking, Quantum Link, 107
 - music, conversion for computers, 246
 - music and sound programming, 235-247
 - musical staff display, 246
- ### N
- naming
 - date and time in, 87
 - programs, 86
 - programs, temporary, 86
 - TI\$ in, 87, 88
 - naming variables, 203
 - NEW command, 15, 18
 - Newsroom. The, 375
 - NEXT, 149
 - NLQ mode, MPS803 printer, 58
 - no-hands game playing, 346
 - noteboard, 29
 - numbering systems, 250
 - numbers, 249
 - large, working with, 260
 - symbols for, 249
 - typing in, 34
 - numeric files, efficient use of, 208
- ### O
- ON clause, 147
 - ON . . . GOTOs, 207
 - One-On-One, 357
 - online information services (see also Quantum Link), 108
 - charge minimizing for, 105
 - encyclopedia, reading practice with, 106
 - online assistance from, 105
 - Quantum Link, 106
 - recording, 104
 - operands, machine language, 169
 - operating systems, CP/M, 197-201
- ### P
- paper
 - color printing quality, 62
 - colored, 63
 - fanfold, homemade, 61
 - fanfold, marking, 61
 - graph, program for, 305
 - heat-sensitive, 62
 - plotter, 66
 - printer-paper alignment tips, 55
 - recycling, 21, 31
 - single sheet feeds for, 61
 - Paperclip, 375
 - Paradox, 139
 - paragraphs, automatic numbering of, 379
 - Paratrooper, 290
 - parenthesis, 9
 - Pascal, 194, 196
 - passwords
 - Quantum Link, 108
 - telecommunications privacy and, 104
 - pauser code, 178, 333
 - PC clones, 132
 - PEEK, 138
 - peripherals
 - inoperative, 117
 - new equipment burn in, 128
 - permutations, 259
 - PETASCII
 - conversions in, 209
 - translating 64 to, 274
 - photographing display screens, 49
 - Piece de Resistance, 283
 - Pitfall, 357
 - place holder, word processing and, 378
 - PLAY statement, 144, 156, 244
 - plotters, paper and pens for, 66
 - PLUG newsletter, 133
 - Plus-4 Computers, 133-135
 - POKE, 138, 155, 161
 - commercial software and, 345
 - F-key disabler using, 43
 - screen displays and, 227
 - polarized connectors, 114
 - POP routine, 212
 - Popeye, 357
 - ports, 113-114
 - Potholes, 277
 - power supplies
 - interference in displays from, 46
 - line filters for, 47
 - printer interface, 52
 - problems in, 118
 - Print Shop, LOGO usage and, 195
 - PRINT@ code, 178, 224
 - printer stand, 27
 - printers, 51-59
 - Blue Chip-MPS802 equivalent, 58
 - bottom-feed, 56
 - cleaning printhead in, 58
 - CP/M, 200
 - DIP switches, 52
 - escape sequence for, 56-57

- extra print modes for, 55
 - fonts, expanding use of, 55
 - hex mode in, 54
 - interface power source, 52
 - label printing, 55
 - labels, gumming up of, 62
 - machine language, 164, 165
 - multiple, 53-54
 - NLQ mode, 58
 - paper alignment, 55
 - pause control for, 54
 - printwheels for, 67-68
 - ribbon compatibility in, 64, 65
 - ribbons for, 64, 65
 - single sheet feeds for, 61
 - software compatibility and, 345
 - supplies for, 60-68
 - SX 64 interface with, 136
 - tractor modification, 57
 - printing, 51-59
 - color, paper quality for, 62
 - labels, 380
 - printwheels, DPS1101, 67-68
 - programmer's reference guides, 3, 4
 - programming
 - BASIC for, 139
 - basics of 137-143
 - duplicating lines in, 36
 - mathematics, 248-261
 - music and sound, 235-247
 - Shifted Return to skip line in, 33
 - video display, 214-234
 - programs
 - 128-Speedup, 15
 - assembler, 138
 - books and magazines containing, 7-19
 - brackets in, 11
 - checksum, typing tips for, 9
 - corrections to published, 10-11
 - crashing, RESET button for, 124
 - database, 139
 - deleting lines from, 212
 - directories and, 142
 - frequently used, automatic
 - loading of, 80
 - graphic symbol key for, 9
 - inoperative, peripheral
 - interference with, 117
 - jamming other programs with, 10
 - keystroke abbreviations and
 - description for listings, 11
 - labeling, 10
 - line number suppression, 212
 - magazine entry, 12, 13
 - menus for, 88
 - naming, 86
 - one-touch running of, 33
 - protection scheme for, 40, 211
 - Quantum Link, 109
 - REM statement removal in, 8
 - serial bus disconnection and, 124
 - slowing down listing of, 33
 - speed of, 96
 - spreadsheet, 139
 - Supermon, 163
 - temporary names for, 86
 - typing in, 8
 - word processing, 139
 - proofreading, 15, 378
 - protection, 40, 211
 - protection keys, 347
 - Psychological Insights, 285
- Q**
- quadruple-density disks, 77
 - Quantum Link, 106
 - BASIC programming and, 140-141
 - multitasking, 107
 - passwords for, 108
 - quick messages on, 107
 - repeating keys for, 108
 - screen names for, 108
 - time saving tips, 106
 - Quest for Tires, 357
 - question mark, INPUT and, 151
 - quick erase, 179, 334
 - quotation mark, 151
 - quote mode, escape sequence, 44
- R**
- Raid Over Moscow, 358
 - RAM
 - expansion units for, 127
 - color, memory locations, 166
 - memory locations, 165, 166
 - Rambo, 358
 - RAMDOS program, 127
 - re-inking ribbons, 66
 - Reach!, 278
 - READ...DATA statements, errors
 - in, 265
 - readers' guide, 3
 - READY prompt, 144
 - reassignment-original assignment
 - of function keys, 41
 - recycling materials, 21, 22, 31, 66
 - reference guides, 3, 4
 - relational expressions, 150
 - REM statements, 8
 - deletion of, 206
 - labeling programs with, 10
 - REMOVED, 179, 333
 - RESET button
 - changing, 213
 - commercial software and, 346
 - disabling, 40
 - extender for, 124
 - installation of, 122
 - program crash and, 124
 - machine language monitor, 163
 - RestoreReset disable, 40
 - uses for, 122, 123
 - RESTORE key, 37-40
 - reverse character mode, escape
 - sequence, 44
 - RGB monitor extension, 49
 - ribbons, 64-66
 - ROM
 - kernel, memory locations, 167
 - memory locations, 166
 - Roman numerals, Arabic
 - conversion for, 255
 - roots, 258
 - rotator, sprite, 231
 - rounding numbers, 257
 - rulers and scales, 29
 - RUN program, 148
 - automatic, 42
 - easy execution of, 90
 - running programs, one-touch
 - keying for, 33
- S**
- S'More Cartridge, 376
 - SAVE, disabling of, 159
 - scaling, symmetrical, 230
 - schematic diagrams, 119
 - scrambler, screen, 219
 - screen bars code, 180, 221
 - screen effect code, 181
 - screen explosion effect, 232
 - screen framer, 220
 - screen mode swapper, 230
 - screen POKES, 227
 - screen saver, 215
 - screen scrambler code, 191, 219
 - screens, cleaning, 30
 - Screentron, 293
 - Sea Battle, 291
 - search code, 181, 335
 - sector editor, 347
 - sectors, 77
 - sensing special keys, 207
 - SEQ files
 - editing, 208
 - reader utility for, 317, 318
 - serial bus, disconnection from, 124, 125
 - Seven Cities of Gold, 358
 - sheet music, conversion for
 - computers, 246
 - SHFT key, 11

Index

- shorting, 117
 - silencer, modem, 103
 - siren sound, 238
 - SIXPACK PLUS program, 169, 183-185, 326-329
 - skipping line of listing, 33
 - slowdown loading problems, 94
 - slowing down display listing, 33
 - software, 202-213
 - 128 keys in 64 mode for, 346
 - bypassing dongles and protection keys in, 347
 - commercial, 344-348, 363-376
 - documentation, 345
 - key-protected, 347
 - POKEs and, 345
 - printer compatibility and, 345
 - RESET button and, 346
 - sales receipts for, 345
 - track and sector editors, 347
 - used vs. new, 23
 - source listings, 169
 - Space Harrier, 358
 - special screen effect program, 225
 - speed, 96, 206
 - speed dialing, 101
 - Speedup 128 program, 15
 - spreadsheets, 139
 - SPREDEF, background colors
 - setting for, 160
 - sprites
 - 128 to 64, 272
 - 64 to 128, 273
 - finder utility for, 326
 - rotator for, 231
 - Spy vs. Spy, 359
 - square roots, 258
 - SSSD disks, 77
 - Star Wars theme music, 242
 - static, 28
 - Stop key, 37-39, 207
 - STOP statement, debugging and, 266
 - storing disks, 82-84
 - STRCMP(), Super C language bug, 196
 - string printer, machine language, 164
 - Summer Games, 359-360
 - Super C language, 196
 - Super Expander, translation of, 272
 - Super Huey, 360
 - Super Pascal, 196
 - Superbase, 139
 - Supermon program, 163
 - surge protection, 28
 - switchbox, TV video displays, 47
 - symbols, numbers and, 249
 - symmetrical scaling, 230
 - syntax errors, 263
 - SYS command, 12, 15, 18
- T**
- TAB statement, 153, 270
 - taxes, computer business-usage and, 22
 - telecommunications, 100-110
 - television set video displays, 46-47
 - TEMPO table maker, 245
 - temporary line deletion, 204
 - temporary program names, 86
 - THEN, 150
 - TI variable, 206
 - TI\$ in filenames, 87, 88
 - timer code, 185, 329
 - Timetrial, 292
 - TPUG, 4
 - track editor, 347
 - tractors, modifications to, 57
 - translating, 271-275
 - 128 and 64 Super Expander programs, 272
 - 128 sprites on 64, 272
 - 64 sprites to 128, 273
 - 64 to 128, 272
 - 64 to PET, 274
 - troubleshooting, 116-120
 - turtle graphics, COMAL and, 195
 - typing, 8
 - duplicate lines, 36
- U**
- Ultima III, 360
 - unseen characters, 264
 - unsticking jammed drives, 93, 94
 - user groups, 2, 4, 5, 6, 23
 - user RAM, memory locations, 166
 - utility programs, 311-344
- V**
- variable dump, 186, 331
 - debugging and, 267
 - variables
 - debugging, 268
 - naming, 203
 - vibrating screen display, 215, 219
 - video monitor switch, 50
 - video test pattern utility, 325
- W**
- Warhawk, 361
 - warranties, enhancing, 22
 - Wedge program (see DOS Wedge)
 - Whiz Bang, 285
 - wind chill calculator, 307
 - Winter Games, 361
 - wiring, 111-115
 - word processing, 139, 377-381
 - automatic paragraph numbering, 379
 - easy programs for, 302-305
 - file organization for, 380
 - formatting for, 378
 - justification, 379
 - placeholder in, 378
 - proofreading, 378
 - unslashing zeros in, 378
 - Worm program, 286
 - write-protection, 30, 78-80, 313
- Z**
- Zaxxon, 361
 - zero, abbreviation of, 34
 - zero values, 207

Other Bestsellers From TAB

TROUBLESHOOTING AND REPAIRING YOUR COMMODORE™ 128—Art Margolls

Hundreds of easy repairs save you time, money, and computer down time! This sourcebook is probably the most important "add-on" you'll ever buy for your C-128! Packed with maintenance tips, troubleshooting procedures, and do-it-yourself repair techniques, it can save you time, money, and untold frustration. The simple chip-changing instructions, alone, will enable you to cure at least 50% of the problems that cause breakdowns. 400 pp., 290 illus.
Paper \$21.95 **Hard \$27.95**
Book No. 3099

COMMODORE 128™ DATA FILE PROGRAMMING—David Miller

Of all programming topics, file handling is the one that most intimidates beginning and intermediate programmers alike. Now, David Miller has developed a guide for C-128 users that takes the misery and mystery out of learning to use the C-128 file structure. After completing this book, you will fully understand what files are and how to use them—plus you'll be able to create your own sequential and relative access files! 292 pp., 12 illus.
Paper \$17.95 **Hard \$21.95**
Book No. 2805

AMIGA™ ASSEMBLY LANGUAGE PROGRAMMING—Jake Commander

Here, machine language specialist and expert program designer Jake Commander provides an in-depth introduction to Commodore's new generation computer, the Amiga, and its sophisticated 68000 microprocessor. He reveals all the assembly language programming techniques you need to make your programming as efficient and effective as possible. If you're a serious programmer, you will welcome this opportunity. 240 pp., 35 illus.
Paper \$15.95 **Hard \$19.95**
Book No. 2711

COMMODORE CARE MANUAL: DIAGNOSING AND MAINTAINING YOUR 64 or 128 SYSTEM—Chris Morrison and Teresa S. Stover

You need no previous experience with computer programming or repair. This book provides you with three basic approaches to Commodore care: preventive maintenance, problem diagnostics, and general repair. You will save time and money! 224 pp., 101 illus.
Paper \$13.95 **Hard \$16.95**
Book No. 3141

COMMODORE 128™ BASIC: Programming Techniques—Martin Hardee

With the help of this guide, you can get unbelievable programming productivity and versatility using BASIC on your C-128! You'll learn disk commands that will allow you to store and retrieve data, telephone communications techniques, and sound and graphic commands that will make your programs clearer and more enjoyable. Includes over 50 practical programs! 190 pp., 120 illus.
Paper \$14.95 **Hard \$18.95**
Book No. 2732

ADVANCED COMMODORE 128™ GRAPHICS AND SOUND PROGRAMMING—Stan Krute

Learning to use your C-128's full sound and graphics potential in three modes is easy with the storehouse of expert tips, advice, and guidance packed into this book. Sample exercises and ready-to-run programs help you to quickly master the fundamentals, and prepare you to start combining sound and graphics for advanced applications. This time-saving reference to C-128 sound and graphics programming highlights a sound and music lab. 400 pp., 72 illus.
Paper \$17.95 **Hard \$21.95**
Book No. 2630

Send \$1 for the new TAB Catalog describing over 1300 titles currently in print and receive a coupon worth \$1 off on your next purchase from TAB.

(In PA, NY, and ME add applicable sales tax. Orders subject to credit approval. Orders outside U.S. must be prepaid with international money orders in U.S. dollars.)
 *Prices subject to change without notice.

To purchase these or any other books from TAB, visit your local bookstore, return this coupon, or call toll-free 1-800-822-8158 (In PA and AK call 1-717-794-2191).

Product No.	Hard or Paper	Title	Quantity	Price

Check or money order enclosed made payable to TAB BOOKS Inc.

Charge my VISA MasterCard American Express

Acct. No. _____ Exp. _____

Signature _____

Please Print

Name _____

Company _____

Address _____

City _____

State _____ Zip _____

Subtotal

Postage/Handling
 (\$5.00 outside
 U.S.A. and Canada)

In PA, NY, and ME
 add applicable
 sales tax

TOTAL

\$2.50

Mail coupon to:

TAB BOOKS Inc.
 Blue Ridge Summit
 PA 17294-0840

LOU SANDER'S TIPS AND TRICKS FOR COMMODORE COMPUTERS

If you are intrigued with the possibilities of the programs included in *Lou Sander's Tips and Tricks for Commodore Computers* (TAB Book No. 3192), you should definitely consider having the ready-to-run disk containing the software applications. This software is guaranteed free of manufacturer's defects. (If you have any problems, return the disk within 30 days, and we'll send you a new one.) Not only will you save the time and effort of typing the programs, but also the disk eliminates the possibility of errors that can prevent the programs from functioning.

Available on disk for the 64 and 128 at \$24.95 for each disk plus \$1.50 shipping and handling.

I'm interested. Send me:

_____ disk(s) (6434S)
_____ TAB BOOKS catalog (\$1.00) (with a coupon worth \$1.00 on your next TAB purchase)

Check/Money Order enclosed for \$24.95 plus \$1.50 shipping and handling for each disk ordered.

VISA MasterCard American Express

Account No. _____ Expires _____

Name _____

Address _____

City _____ State _____ Zip _____

Signature _____

Mail To: **TAB BOOKS Inc.**

Blue Ridge Summit, PA 17294-0850

OR CALL TOLL-FREE TODAY: 1-800-822-8158

IN PENNSYLVANIA AND ALASKA CALL: 717-794-2191

(In PA, NY, and ME add applicable sales tax. Orders subject to credit approval. Orders outside U.S. must be prepaid with international money orders in U.S. dollars.)

*Prices subject to change without notice.

TAB 3192

**See page 391 for a Special
Companion Diskette Offer.**

I'm interested. Send me:

_____ disk(s) (6434S)
_____ TAB BOOKS catalog (\$1.00) (with a coupon worth \$1.00 on your next TAB purchase)

Charge my credit card for _____ (\$24.95 plus \$2.50 shipping and handling for each disk ordered).

Check one: VISA MasterCard American Express

Account No. _____ Expires _____

Signature _____

Name _____

Address _____

City _____ State _____ Zip _____

**OR CALL TOLL-FREE TODAY: 1-800-822-8158
IN PENNSYLVANIA AND ALASKA, CALL: 717-794-2191**

*Prices subject to change without notice.

(In PA, NY, and ME add applicable sales tax. Orders subject to credit approval. Orders outside U.S. must be prepaid with international money orders in U.S. dollars.)

TAB 3192



**NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES**

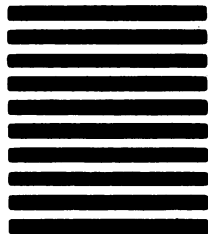
BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 9 BLUE RIDGE SUMMIT, PA 17214

POSTAGE WILL BE PAID BY ADDRESSEE

TAB BOOKS Inc.

Blue Ridge Summit, PA 17214-9988



LOU SANDER'S TIPS AND TRICKS FOR COMMODORE™ COMPUTERS

LOUIS F. SANDER

"A nice collection of hints and small programs, covering a wide range of subject areas." —**Jim Butterfield**, noted writer, speaker, and consultant

"... a mixture of the clever tips and down-to-earth advice that mother would have given you if mother used a Commodore. This book offers the quickest possible route to expertise." —**Barbara Mintz**, Boston Computer Society

"A potpourri of programs, a plethora of game tips and a pile of useful tricks make (this book) both good reading and an indispensable reference tool for any Commodore computer owner." —**James Gracely**, Managing Editor
Commodore Magazine

Here is the perfect source of ideas and expert guidance for getting the most out of your Commodore! Well-known Commodore authority Louis F. Sander has put together this compendium of over 600 hints and bits of advice taken from his popular *Commodore Magazine* column "Tips and Tricks." It includes valuable information on virtually every aspect of computing, from simple hardware tips to machine language programming to designing your computer room, and includes over 150 programs.

A sampling of the many areas covered includes:

- Telecommunications
- Troubleshooting
- Programming music and sound
- Applications programs
- Video displays
- Machine language
- Utility programs
- Word processing

Each of the included items was contributed by a computerist just like you. This is the only book of its kind for Commodore users, and certainly the most complete, useful guide you'll find.



WINDCREST

Blue Ridge Summit, PA 17294-0850

0489

\$21.95

ISBN 0-8306-3192-5



9 780830 631926