

Commodore 64™ Programs for the Home

Charles D. Sternberg



HAYDEN

COMMODORE 64™ PROGRAMS
for the HOME

COMMODORE 64™ PROGRAMS

for the HOME

CHARLES D. STERNBERG



HAYDEN BOOK COMPANY
a division of Hayden Publishing Company, Inc.
Hasbrouck Heights, New Jersey

Production Editor: TERRY DONOVAN
Art Director: JIM BERNARD
Production Service: THE BOOKMAKERS, INCORPORATED
Printed and bound by: ARCATA GRAPHICS CO.; FAIRFIELD
GRAPHICS DIVISION

Commodore 64™ is a trademark of Commodore Business Machines, Inc.

Copyright © 1983 by HAYDEN BOOK COMPANY, INC. All rights reserved. No part of this book may be reprinted, or reproduced, or utilized in any form or by any electronic, mechanical, or other means, now known or hereafter invented, including photocopying and recording, or in any information storage and retrieval system, without permission in writing from the Publisher.

Printed in the United States of America

2	3	4	5	6	7	8	9	PRINTING
84	85	86	87	88	89	90	91	YEAR

Equipment Needed

To use the programs in this book you will need the following equipment:

- Commodore 64 microcomputer
- Television set (B/W or color)
- Commodore Datasette
- Commodore VIC-1441 Disk Drive (optional)

All the programs in this book are designed to be run using the datasette. If you wish to store or use any of these programs on diskette, Appendix B will help you make the necessary changes to the program listings.

Contents

1	Introduction	1
2	Financial	5
	MORTGAGE	Computing mortgage payments 5
	PAYMENTS	Projecting monthly payments and repayment schedules 8
	INTEREST	Projecting interest on savings 11
	UTILITY	Analyzing utility usage 13
	BUDGET	Household payment and budget projections 16
3	Household Helpers	19
	EVENTREC	Recording and retrieval of events 19
	TELEDIR	Telephone directory recall 23
	GIFTPLAN	Gift planning and cost projections 26
	MEALPLAN	Meal planning and grocery list creation 28
	WEIGHT	Weight loss recording and analysis 32
4	Other Household Helpers	36
	CHORES	Family chore scheduling 36
	TASKS	Recording and retrieval of required activities 41
	CALC	Turning a Commodore into a calculator 44
	CAR-REC	Car maintenance recording and retrieval 47
	JOB-EST	Cost estimation for home projects 51
5	For the Home Worker	56
	WORK-REC	Work and vacation time recording and reporting 56
	PROFLOSS	Profit and loss calculation program 60
	ORDER-FM	Order entry for the home businessperson 63
	ORDER-PR	Order retrieval for the home businessperson 70
	BIDCREAT	Assistance in bid creation and preparation 73
6	Children's Work and Play	80
	MPRINT-1	Multiple printing program from keyboard entries 80
	MPRINT-2	Multiple printing program from data statements 82
	EXPENSE	Calculates a child's budget/expenses 85
	CODER	Produces coded/decoded messages 89
	N-PAPER	Calculates cost/earnings for newspaper delivery 94

7	Schoolwork	97
	FLASH-CD Produces arithmetic flash cards	97
	W-SOUNDS Assists beginning readers	106
	OUTLINE Creates standard outlines from input data	110
	QA-BUILD Builds question-and-answer files for homework/test review	117
	QA-TUTOR Presents questions and evaluates responses for homework/test review	121
8	Outside Activities and Recreation	125
	TRIPPLAN Trip planning and cost estimation	125
	DICEROLL Rolls dice to the screen	129
	FOOTBALL A program for win/lose projections	132
	SCORE-BD Computes league standings and prints schedules	139
	DUESLIST Records and prints records of dues owed and dues paid for club activities	146
9	For the Commodore 64 User	148
	BUILD-TF Build tape (or disk) files from keyboard data	148
	PRINT-TF Prints sequential tape (or disk) files	152
	TAPE-LIB Records and prints directories of file and program storage locations	155
	HEX-DEC Converts hexadecimal numbers to their decimal equivalents	159
	COLUMN General purpose report-generation program	161
	Appendix A Language Features Used	166
	Appendix B Converting Programs for Floppy Disk Usage	168

COMMODORE 64™ PROGRAMS
for the HOME

1 Introduction

Your Commodore 64 is a useful machine — assisting you and your family with a wide variety of household tasks and activities. The programs in this book provide a good cross-section of practical applications that will make your computer beneficial to all family members.

The individual applications have been designed with simplicity in mind, to make them as valuable as possible to the first-time computer user. Further, this simple approach to the design and presentation of each program and application allows you to modify these programs for other uses as you gain familiarity with writing programs for your Commodore 64. Until that time, however, your computer can be establishing itself as a useful and productive part of your household.

The Book's Format

Each of the programs presented in this book has been designed, run, and tested on the Commodore 64 computer. The individual programs can be entered directly from the program listings provided in the following chapters. Without any knowledge beyond what is presented in this book, you can operate these programs to perform their selected tasks for your household. To assist you, each program has been provided in such a way as to facilitate your understanding and use of it in your home. Each program is presented in the following form:

1. A narrative description of the program, application, function, and usefulness.
2. An explanation of how to use the program.
3. A description of how data are to be entered for the program's use.
4. A description of items to check before you run the program or if you have problems with its operation.
5. An explanation of how the program works. This section is designed to improve your understanding of the programming process and to

provide you with the information you need to modify or change the program's operation.

6. A complete listing of the program — ready for entry into your Commodore 64.
7. Sample runs from the program to illustrate actual data and output results. Whenever possible, all options are shown. These data can be used to familiarize yourself with the program's use. They provide a double check of the program after it is entered into your computer.
8. A table of all major variable names to use in modifying and understanding each program. A list of the less frequently used statements and functions is also provided.

Entering and Using the Programs

The programs can be used exactly as they are shown (see note directly preceding the Mortgage Program.) Once you enter them into your machine's memory, you can use them to perform the tasks they are designed to carry out. The following sequence of steps should be followed to perform each task:

1. First, note the difference between the numeric zero and the alphabetic O, as they appear in the sample listing. While in some cases there is only a slight difference to the eye, there is a major difference to your computer. Note also the difference between the numeric one and the alphabetic I, since the same comment applies to these characters.

2. Using the keyboard, enter each statement exactly as it appears in the listing. If you make an error during input, you can use the cursor control keys (CRSR) and/or the insert/delete (INST DEL) key to make the necessary change to the line. If the error is too severe to make this method easy, however you can replace the entire line by reentering it later. Do not forget to press the carriage return (RETURN) key after each line is entered, since the line is not entered into the machine's memory until it sees RETURN.

3. After entering all the program lines, you should double-check each one to ensure that you have not mis-typed an item. You can do this by using the LIST command. Typing LIST will cause the computer to display on its screen all the lines in its memory. If the program is very long, you will find that you will not be able to keep up with the rapid movement of the listing as it is displayed. You can either slow the listing down by holding the control (CTRL) key down while the listing is being displayed, or you can list only specific portions of the program. To check the program properly, I recommend that you list it ten lines at a time by entering LIST 1-100 (for the first ten lines), then LIST 101-200 (for the next ten), and so on. If you find an error through this process, you can

correct it either by reentering the line correctly at the bottom of the screen or by moving the cursor (using the CRSR key) to the error and then using the INST DEL key or typing over the erroneous item. Do not forget to press the RETURN key before you move the cursor from that line. If you neglect to press RETURN, the change will not be recorded. After making any necessary corrections, you should simultaneously press the CTRL key and the CLR HOME key to clear the screen. You should then relist the same sequence of lines to make sure that you made all your corrections properly.

4. After checking your program, you should try running it using the sample data. Your result should be an exact duplicate of the sample outputs that are shown following the program listings. If the results differ, you should recheck your program to determine where it does not match the listing from the book. Correct your errors in the manner explained above.

5. Once you have tested your program and produced the expected results, you should continue to run it using sample data of your own. Doing this will help you to understand the operation of the program and learn how to enter data for it.

6. Once you are sure the program has been entered correctly, you should save it for later use. If you have a Commodore disk, you can accomplish this by using the command SAVE "program name," then 8. The program you have entered into memory will then be saved on the disk under the name you have indicated as program name. If you are using the cassette recorder for storage, you should position the tape to the proper location and then enter SAVE "program name". The screen will then tell you to PRESS RECORD & PLAY ON TAPE. After you press the RECORD & PLAY keys, the screen will go blank for a few seconds. Then it will redisplay the previous data with the word OK following the previous line. Before you use these devices, however, you should have reviewed their operation in the appropriate Commodore User's Manual.

Compatibility of Language Features

Although the programs in this book represent specific applications for use on the Commodore 64 computer, they should not create too many difficulties for users of other computer systems. Appendix A presents specific information on Commodore's instruction formats as they are used in the programs. In addition, the symbol table provided with each program contains special statements/functions that may differ from those of other computer systems.

Program Efficiency

These programs have purposely been provided in a form that does not minimize either memory usage or processing speed. Providing this efficiency would have made the programs much less easy for you to read and much more difficult for you to modify for other uses. You can increase the programs' efficiency in the following ways, but these changes are not recommended. The programs are so small that the impact of changing them would not be significant.

Line numbers can be changed from their present interval of ten to intervals of one or two. Smaller line numbers use smaller amounts of memory.

Multiple instructions can be placed on a single line separated by a semicolon. If you do this though, watch out for IF and GOTO statements.

REM statements can be removed from the program.

Spaces can be removed from the program.

Note that although each of these actions will reduce the program's memory requirements, they all do so at the expense of readability and human understanding. They should be accomplished only if they are absolutely required.

2 Financial

MORTGAGE

Description

This program accepts a mortgage amount, interest rate, and number of years for the mortgage from the keyboard and computes a monthly payment of principal and interest based upon those entries. It can be used to look at various mortgage and refinancing alternatives.

Instructions for Use

RUN the program and then respond to the questions asked.

Data Entry

All data are entered in response to the program's messages.

Notes and Things to Check

Internal computer storage of numbers is sometimes an approximation of the actual number. Even though this approximation is extremely close, some minor differences may occur in results. The monthly payment that is shown may therefore differ slightly from actual payment amounts.

Modification Procedures

The calculation of the payment occurs on lines 180 and 190. The only other calculation occurs at lines 100-110, where the interest rate is checked to determine if it was entered as a whole number or as a decimal.

Note to All Programs

The Commodore 64 has a screen width of forty characters. This means that you can have only forty letters and spaces across each line on

the screen at one time. When you reach forty characters on a line, the next character will automatically shift to the beginning of the next line.

The programs in this book have line lengths of up to seventy characters. This has been done so they can be read and typed from the book with greater ease and efficiency. When you type a line from the book that is more than forty characters, the line will automatically "wrap around" to the next line on your screen at the forty-first character. Let your computer do this. Hit the RETURN key only when it is time to start typing a new line with a new line number. Each time you hit the RETURN key you are sending a signal to your computer to look for the next line number.

```
10 REM          SAVED AT MORTGAGE
20 REM PROGRAM TO FIND MORTGAGE PAYMENT
30 REM PRINCIPAL AND INTEREST ONLY
40 REM *****
50 REM          DATA INITIALIZATION
60 REM *****
70 INPUT "ENTER THE MORTGAGE AMOUNT";P
80 INPUT "ENTER THE INTEREST RATE";I1
90 IF I1>1 THEN 110
100 I1=I1*100
110 I=(I1/100)/12
120 INPUT "ENTER MORTGAGE YEARS";Y
130 PRINT
140 PRINT
150 REM *****
160 REM          COMPUTATION AND PRINT
170 REM *****
180 M=I/((1+I)^(Y*12)-1)+I
190 M1=M*P
200 PRINT "*****"
210 PRINT "MORTGAGE AMOUNT $";P
220 PRINT "INTEREST RATE ";I1;"%"
230 PRINT "MONTHLY PAYMENT $";M1
240 PRINT "*****"
250 REM *****
260 REM          PROGRAM TERMINATION POINT
270 REM *****
280 PRINT
290 PRINT
300 STOP
```

RUN

ENTER THE MORTGAGE AMOUNT? 10000
ENTER THE INTEREST RATE? 12
ENTER MORTGAGE YEARS? 10

MORTGAGE AMOUNT \$ 10000
INTEREST RATE 12 %
MONTHLY PAYMENT \$ 143.470949

Break in 300

MAJOR SYMBOL TABLE — MORTGAGE

NAME	DESCRIPTION
I	MONTHLY INTEREST
I1	INTEREST RATE
M	COMPUTATION VARIABLE
M1	MONTHLY PAYMENT
P	MORTGAGE AMOUNT
Y	YEARS FOR THE MORTGAGE

FUNCTIONS USED

NAME
NONE

PAYMENTS

Description

This program projects monthly payments and repayment schedules for installment buying. The user is given the choice of whether or not to see the repayment schedule. This program can be used to evaluate the timing of purchases and/or alternatives.

Instructions for Use

RUN the program and then respond to the program's request for amount to be borrowed, interest rate, and number of payments to be made.

Data Entry

All data are entered in response to program messages.

Notes and Things to Check

Internal storage methods for numbers may create small differences in projected payment amounts and repayment schedules.

Modification Procedures

Line #160 computes the payment amount. Lines 240-340 calculate and print the repayment schedule, with the calculation of interest paid at line #300 and balance owed at line #320. Note that the INT function is used in the calculation to show only two decimal positions.

```
10 REM          SAVED AT PAYMENTS
20 REM PROGRAM COMPUTES PAYMENT AMOUNT
30 REM AND OPTIONAL PAYMENT SCHEDULE
40 X$="*****"
50 C0=.5
60 INPUT "ENTER AMOUNT BORROWED";D
70 INPUT "      INTEREST RATE";R
80 IF R>1 THEN R=R/100
90 C=12
100 INPUT "      NUMBER OF PAYMENTS TO BE MADE";N
110 R1=R/C
120 PRINT
130 REM *****
140 REM          PROCESSING AREA
150 REM *****
```

```

160 P=INT(C0+D*(R1/(1-(1+R1)^(-N)))*100)/100
170 PRINT "PAYMENT (PRINC AND INT) IS: ";P
180 PRINT
190 PRINT "WOULD YOU LIKE TO SEE A REPAY SCHEDULE (Y OR N)"
200 INPUT A$
210 IF A$<>"Y" THEN 370
220 PRINT X$
230 PRINT
240 PRINT TAB(10);"AMORTIZATION SCHEDULE"
250 PRINT
260 PRINT "PER";TAB(5);"PAYMENT";TAB(14);"INT";TAB(22);
270 PRINT "TO PRINC.";TAB(32);"BAL"
280 PRINT TAB(31);D
290 FOR I=1 TO N
300 I1=INT(C0+D*R1*100)/100
310 PRINT I;TAB(5);P;TAB(14);I1;TAB(23);P-I1;
320 D=INT(C0+(D-(P-I1))*100)/100
330 PRINT TAB(30);D
340 NEXT I
350 PRINT X$
360 REM *****
370 REM PROGRAM TERMINATION POINT
380 REM *****
390 PRINT
400 PRINT
410 PRINT "PROCESSING COMPLETE"
420 PRINT
430 STOP

```

```

RUN
ENTER AMOUNT BORROWED? 1000
INTEREST RATE? 18
NUMBER OF PAYMENTS TO BE MADE? 12

PAYMENT (PRINC AND INT) IS: 91.68

WOULD YOU LIKE TO SEE A REPAY SCHEDULE
(Y OR N)
? Y

```

AMORTIZATION SCHEDULE

PER	PAYMENT	INT	TO PRINC.	BAL
				1000
1	91.68	15	76.68	923.32
2	91.68	13.85	77.83	845.49
3	91.68	12.68	79	766.49
4	91.68	11.5	80.18	686.31
5	91.68	10.29	81.39	604.92
6	91.68	9.07	82.61	522.31
7	91.68	7.83	83.85	438.46
8	91.68	6.58	85.1	353.36
9	91.68	5.3	86.38	266.98
10	91.68	4	87.68	179.3
11	91.68	2.69	88.99	90.31
12	91.68	1.35	90.33	-.02

PROCESSING COMPLETE

Break in 430

MAJOR SYMBOL TABLE — PAYMENTS

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
C	# OF COMPOUNDING PERIODS/YEAR
CO	ROUNDING CONSTANT
D	AMOUNT OWED
I	INDEX AND ARRAY POINTER
I1	INTEREST PAID DURING PERIOD
N	NUMBER OF PAYMENTS
P	PAYMENT AMOUNT
R	INTEREST RATE
R1	INTEREST RATE PER PERIOD

FUNCTIONS USED

NAME
INT
TAB

INTEREST

Description

This program computes daily compounded interest. It can be used to project interest to be earned on savings accounts or other investments.

Instructions for Use

RUN the program and provide the information requested by the program.

Data Entry

All data are entered through the keyboard in response to the program's requests.

Notes and Things to Check

The internal storage of numbers by the computer may cause the results of this program to differ slightly from the actual numbers computed by your bank. The results of this program should therefore be considered an approximation of the actual interest that will accrue.

Modification Procedures

Computing interest compounded daily requires a new calculation for each day. This calculation is accomplished by lines 170-210 in the program. The print-out of results does not occur until the last day has been calculated. The total amounts printed (T and T5) are rounded to dollars and cents at lines #230 and #240 of the program.

```
10 REM      SAVED AT INTEREST
20 REM THIS PROGRAM COMPUTES DAILY COMPOUNDED INTEREST
30 REM *****
40 C=365
50 INPUT "ENTER AMOUNT IN SAVINGS";T0
60 INPUT "      # OF DAYS TO PROJECT";N
70 INPUT "      INTEREST RATE";R
80 X$="*****"
90 PRINT X$
100 PRINT "      AMOUNT IN SAVINGS";T0
110 PRINT " ANNUAL INTEREST RATE";R;"%"
120 PRINT "      COMPOUNDED DAILY"
130 PRINT X$
140 R1=R*.01/C
```

```

150 T=T0
160 T5=0
170 FOR K=1 TO N
180 T1=T*R1
190 T5=T5+T1
200 T=T+T1
210 NEXT K
220 PRINT "BEGIN";TAB(10);"INTEREST";TAB(22);"TOTAL"
230 T=INT((T+5/1000)*100)/100
240 T5=INT((T5+5/1000)*100)/100
250 PRINT T0;TAB(10);T5;TAB(22);T
260 REM *****
270 REM PROGRAM TERMINATION POINT
280 REM *****
290 PRINT
300 STOP

```

```

RUN
ENTER AMOUNT IN SAVINGS? 1000
# OF DAYS TO PROJECT? 180
INTEREST RATE? 13.25
*****
AMOUNT IN SAVINGS 1000
ANNUAL INTEREST RATE 13.25 %
COMPOUNDED DAILY
*****
BEGIN INTEREST TOTAL
1000 67.51 1067.51

Break in 300

```

MAJOR SYMBOL TABLE — INTEREST

NAME	DESCRIPTION
C	COMPOUNDING PERIOD
K	INDEX POINTER
R1	DAILY INTEREST RATE
T	TOTAL SAVINGS
T0	INITIAL SAVINGS
T5	TOTAL INTEREST
X\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
INT
TAB

UTILITY

Description

This program analyzes and prints utility usage information from meter readings that have been entered using data statements. It can be used to project your monthly utility bills and to assist you in utility conservation efforts.

Instructions for Use

The utility cost/unit should be determined and entered as the first data statement. Each day on which you read the meter, the data you obtain should be entered as the last data statement in the program. You do not need to read the meter every day, but you must read it on the last day of each month and the first day of the following month for the program to operate correctly.

Data Entry

All data are entered using DATA statements. The first DATA statement must contain the utility cost for each unit shown on the meter. Subsequent entries must contain the day of the month and the meter reading, separated by a comma.

Notes and Things to Check

If the program gives erroneous results, you should check to make sure that each data statement contains the required number of entries and that a comma is used to separate them. In addition, the program requires an entry for both the last day of the month and the first day of the following month to insure that the number of days and the average daily usage are correct.

Modification Procedures

Lines 80 and 90 read the cost per unit and the initial meter-reading information. The main calculation is accomplished in the loop at lines 180-340. Average usage and costs are not calculated until lines 440 and 450 are executed.

```
10 REM          SAVED AT UTILITY
20 REM PROGRAM TO ANALYZE UTILITY USAGE
30 REM *****
40 M=1000
```

```

50 PRINT
60 PRINT
70 PRINT
80 READ C
90 READ D,S
100 PRINT "INITIAL READING WAS";S;"ON DAY";D
110 PRINT "COST PER UNIT IS ";C
120 PRINT
130 PRINT "DATE"
140 PRINT "READ";TAB(7);"READ";TAB(16);"USED";TAB(22);" COST"
150 PRINT "-----";TAB(7);"-----";TAB(15);"-----";TAB(22);"-----"
160 T2=S
170 D2=D
180 FOR I=1 TO M
190 N=1
200 READ D
210 IF D=0 THEN 350
220 IF D<D2 THEN 240
230 N=D-D2
240 D2=D
250 READ R
260 T0=R-T2
270 T1=T0*C
280 PRINT D;TAB(5);R;TAB(15);T0;TAB(22);T1;
290 IF N<>1 THEN PRINT "(";N;"DAYS)";
300 T2=R
310 T3=T3+T1
320 N1=N1+N
330 PRINT
340 NEXT I
350 REM *****
360 REM PROGRAM TERMINATION POINT
370 REM *****
380 T4=R-S
390 PRINT TAB(15);"-----";TAB(22);"-----"
400 PRINT " TOTALS";TAB(15);T4;TAB(22);T3
410 PRINT
420 PRINT "*****"
430 PRINT "FOR ";N1;"DAYS"
440 PRINT "AVG DAILY USE WAS:";T4/N1
450 PRINT "AVG DAILY COST WAS:";T3/N1
460 PRINT "*****"
470 PRINT
480 PRINT
490 STOP
500 REM *****
510 REM DATA IS ENTERED HERE
520 REM *****
530 DATA .025
540 DATA 24,1500
550 DATA 25,1590
560 DATA 26,1700
570 DATA 27,1800
580 DATA 28,2200
590 DATA 30,2600
600 DATA 1,3000
610 DATA 2,3100

```

620 DATA 3,3200
 630 DATA 5,3500
 640 DATA 0

RUN

INITIAL READING WAS 1500 ON DAY 24
 COST PER UNIT IS .025

DATE	READ	READ	USED	COST
	25	1590	90	2.25
	26	1700	110	2.75
	27	1800	100	2.5
	28	2200	400	10
	30	2600	400	10 (2 DAYS)
	1	3000	400	10
	2	3100	100	2.5
	3	3200	100	2.5
	5	3500	300	7.5 (2 DAYS)
TOTALS			2000	50

 FOR 11 DAYS
 AVG DAILY USE WAS: 181.818182
 AVG DAILY COST WAS: 4.545455

Break in 490

MAJOR SYMBOL TABLE – UTILITY

NAME	DESCRIPTION
C	COST PER UNIT
D	INITIAL READING DAY
D2	LAST DAY READ
I	INDEX POINTER
M	MAXIMUM NUMBER OF DATA ITEMS
N1	DAY COUNT
R	READING
S	INITIAL READING
T0	UNITS OF SERVICE USED
T1	DAILY COST
T2	LAST READING
T3	TOTAL COST
T4	TOTAL USE

FUNCTIONS USED

NAME
TAB

BUDGET

Description

This program produces projected monthly expenses and budgets based upon data entered via DATA statements. It can be used to assist you in household budgeting and financial decision making.

Instructions for Use

Enter projected monthly income at line #540, then enter as many data statements as you require for monthly bills. An asterisk cues the program to start looking for expenses, which are then entered, terminated by a DATA END entry. After these lines have been entered, the program can be run at any time during the month. For subsequent months, only the changed items need to be corrected to produce accurate results.

Data Entry

Data are entered in three areas: income, bills, and expenses. Income is the first DATA entry. The next entries represent monthly bills. Each bill must show the bill name, followed by the name of the company to be paid, followed by the monthly amount — each separated by a comma. A DATA statement containing an asterisk ends this section.

The next section contains monthly expense information. Each item is entered as a DATA statement, with the expense name, followed by the projected amount (separated by a comma). A DATA statement containing END terminates this section and all other data entry for the program.

Notes and Things to Check

If the program fails to operate correctly, it is likely that the data have been entered incorrectly. This usually occurs through the addition or omission of a required entry or of a comma. Lack of an asterisk following the bills or of a DATA END entry will also cause incorrect data results.

Modification Procedures

Line #190 reads income, lines 200–250 read bills, and lines 270–320 read other expense information. Note that the data input “GET DATA” and the printing of results “PRINT RESULTS” are accomplished by using GOSUBs from the main program that ends at line #150.

```

10 REM          SAVED AT BUDGET
20 REM THIS PROGRAM ALLOWS HOUSEHOLD
30 REM BILLS AND BUDGET PROJECTIONS
40 REM *****
50 M=100
60 DIM B$(M),C$(M),D$(M),A(M)
70 GOSUB 170          REM GET DATA
80 GOSUB 360          REM PRINT RESULTS
90 REM *****
100 REM PROGRAM TERMINATION POINT
110 REM *****
120 PRINT
130 PRINT "PROCESSING COMPLETE"
140 PRINT
150 STOP
160 REM *****
170 REM          GET DATA
180 REM *****
190 READ S
200 FOR I=1 TO M
210 READ B$(I)
220 IF B$(I)="*" THEN 260
230 READ C$(I),D$(I),A(I)
240 T=T+A(I)
250 NEXT I
260 M1=I
270 FOR I=M1 TO M
280 READ B$(I)
290 IF B$(I)="END" THEN 330
300 READ A(I)
310 T=T+A(I)
320 NEXT I
330 M2=I-1
340 RETURN
350 REM *****
360 REM          PRINT RESULTS
370 REM *****
380 PRINT "BILL";TAB(14);"OWED";TAB(30);"DUE";TAB(34);"AMT"
390 PRINT "----";TAB(14);"----";TAB(30);"----";TAB(34);"----"
400 FOR I=1 TO M1-1
410 PRINT B$(I);TAB(14);C$(I);TAB(30);D$(I);TAB(34);A(I)
420 NEXT I
430 FOR I=M1 TO M2
440 PRINT B$(I);TAB(34);A(I)
450 NEXT I
460 PRINT TAB(32);"-----"
470 PRINT "TOTAL BILLS";TAB(33);T
480 PRINT "    INCOME";TAB(30);S
490 PRINT "TOTAL DISPOSABLE INCOME";TAB(34);S-T
500 RETURN
510 REM *****
520 REM          DATA ENTRY FOLLOWS
530 REM *****
540 DATA 1800
550 DATA CAR,XYZ BANK,1,146
560 DATA HOUSE RENT,AAA REALTY,1,450
570 DATA CREDIT CARD,XYZ BANK,15,25

```

```

580 DATA CREDIT CARD,RETAILER Y,24,30
590 DATA *
600 DATA ELECTRIC,95
610 DATA WATER,10
620 DATA TELEPHONE,23
630 DATA FOOD,300
640 DATA GAS,25
650 DATA CLOTHING,50
660 DATA ENTERTAINMENT,30
670 DATA MISC,50
680 DATA END

```

RUN

BILL	OWED	DUE AMT	
----	-----	---	----
CAR	XYZ BANK	1	146
HOUSE RENT	AAA REALTY	1	450
CREDIT CARD	XYZ BANK	15	25
CREDIT CARD	RETAILER Y	24	30
ELECTRIC			95
WATER			10
TELEPHONE			23
FOOD			300
GAS			25
CLOTHING			50
ENTERTAINMENT			30
MISC			50
TOTAL BILLS			1234
INCOME		1800	
TOTAL DISPOSABLE INCOME			566

PROCESSING COMPLETE

Break in 150

MAJOR SYMBOL TABLE — BUDGET

NAME	DESCRIPTION
A()	PAYMENT AMOUNT ARRAY
B\$()	BILL NAME ARRAY
C\$()	COMPANY OWED ARRAY
D\$()	DATE DUE ARRAY
I	INDEX AND ARRAY POINTER
M	MAXIMUM ENTRIES ALLOWED
M1	NUMBER OF BILLS FOUND
M2	NUMBER OF OTHER EXPENSES FOUND
S	INCOME
T	TOTAL EXPENSES

FUNCTIONS USED

NAME
DIM
GOSUB
RETURN
TAB

3 Household Helpers

EVENTREC

Description

This program assists you to record and recall important events. Because you are allowed to record each event under a category, you can review selected categories of information whenever you like. The program can be used to record everything from birthdays to medical occurrences.

Instructions for Use

Each item to be recorded must be entered as a series of DATA statements, and the program must then be resaved to disk or tape with the new data entries. The program can then be RUN, and either all entries or just those you select can be printed.

Data Entry

Data are entered to the program using DATA statements only. Each line (from 640 to 670) in the sample data is entered and saved for future use. The form of each item is shown at line #630 as month, day, type of event, and the event itself. Note that line #630 must remain as the first data item and the DATA END statement (line #680) must always be the last statement.

Notes and Things to Check

If the program fails to function correctly, you should check the DATA statements for completeness and make sure you have used the comma separator whenever it is required. Incorrect data entry is clearly indicated when data appear under inappropriate column headings in a print of all records.

Modification Procedures

Program size can be expanded or contracted by changing the value of M at line #70. Because this variable sets the array sizes, it needs only to be larger than the number of data items to be recorded. Note that the print of all items is accomplished by transferring control to the subroutines at line 380 or line 490.

```
10 REM      SAVED AT EVENTREC
20 REM THIS PROGRAM RECORDS VARIOUS
30 REM EVENTS USING DATA STATEMENTS
40 REM AND THEN LATER SELECTIVELY
50 REM RECALL THOSE EVENTS
60 REM *****
70 M=1000
80 N=4
90 DIM S$(N),H$(N)
100 IF N=0 THEN 310
110 FOR K=1 TO N
120 READ H$(K)
130 NEXT K
140 PRINT "SHALL I PRINT ALL OF THE ITEMS (Y OR N)":INPUT A$
150 IF A$="Y" THEN 220
160 PRINT "ENTER THE ITEM TO SEARCH"
170 INPUT "MONTH (M) OR RECORD TYPE (T)";X1$
180 K1=1
190 IF X1$="T" THEN K1=3
200 PRINT "ENTER THE VALUE OF ";H$(K1);" TO PRINT";
210 INPUT X$
220 PRINT
230 PRINT
240 PRINT
250 PRINT H$(1);TAB(7);H$(2);TAB(12);H$(3);TAB(20);H$(4)
260 PRINT "-----";TAB(7);"-----";TAB(12);"-----";TAB(20);"-----"
-----"
270 PRINT
280 IF A$="N" THEN GOSUB 490          'PRINT SELECTED ITEMS
290 IF A$="Y" THEN GOSUB 380      'PRINT ALL ITEMS
300 REM *****
310 REM PROGRAM TERMINATION POINT
320 REM *****
330 PRINT
340 PRINT "PROCESSING COMPLETE"
350 PRINT
360 STOP
370 REM *****
380 REM      PRINT ALL ITEMS
390 REM *****
400 FOR I=1 TO M
410 FOR K=1 TO N
420 READ S$(K)
430 IF S$(1)="END" THEN 470
440 NEXT K
450 PRINT S$(1);TAB(7);S$(2);TAB(12);S$(3);TAB(20);S$(4)
```



```

460 NEXT I
470 RETURN
480 REM *****
490 REM     PRINT SELECTED ITEMS
500 REM *****
510 FOR I=1 TO M
520 FOR K=1 TO N
530 READ S$(K)
540 IF S$(1)="END" THEN 590
550 NEXT K
560 IF X$<>S$(K1) THEN 580
570 PRINT S$(1);TAB(7);S$(2);TAB(12);S$(3);TAB(20);S$(4)
580 NEXT I
590 RETURN
600 REM *****
610 REM     DATA ENTRY FOLLOWS
620 REM *****
630 DATA MONTH,DAY,TYPE,EVENT
640 DATA JUN,1,HEALTH,JIM-TETANUS SHOT
650 DATA JUN,23,SCHOOL,JAMIE GRADUATED 8TH
660 DATA JUL,6,WORK,PROMOTED TO LVL 3
670 DATA JUL,8,HEALTH,JAMIE 3 STITCHES
680 DATA END

```

```

RUN
SHALL I PRINT ALL OF THE ITEMS (Y OR N)
? Y

```

MONTH	DAY	TYPE	EVENT
JUN	1	HEALTH	JIM-TETANUS SHOT
JUN	23	SCHOOL	JAMIE GRADUATED 8TH
JUL	6	WORK	PROMOTED TO LVL 3
JUL	8	HEALTH	JAMIE 3 STITCHES

PROCESSING COMPLETE

Break in 360

```

RUN
SHALL I PRINT ALL OF THE ITEMS (Y OR N)
? N
ENTER THE ITEM TO SEARCH
MONTH (M) OR RECORD TYPE (T)? M
ENTER THE VALUE OF MONTH TO PRINT? JUL

```

```

MONTH DAY  TYPE      EVENT
-----
JUL   6   WORK      PROMOTED TO LVL 3
JUL   8   HEALTH    JAMIE 3 STITCHES

```

PROCESSING COMPLETE

Break in 360

```

RUN
SHALL I PRINT ALL OF THE ITEMS (Y OR N)
? N
ENTER THE ITEM TO SEARCH
MONTH (M) OR RECORD TYPE (T)? T
ENTER THE VALUE OF TYPE TO PRINT? HEALTH

```

```

MONTH DAY  TYPE      EVENT
-----
JUN   1   HEALTH    JIM-TETANUS SHOT
JUL   8   HEALTH    JAMIE 3 STITCHES

```

PROCESSING COMPLETE

Break in 360

MAJOR SYMBOL TABLE — EVENTREC

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
H\$()	ITEM HEADING ARRAY
K	INDEX AND ARRAY POINTER
K1	POINTER TO SEARCH FIELD
M	MAXIMUM ARRAY SIZE
N	NUMBER OF DATA ITEMS IN EACH RECORD
S\$()	DATA STORED ARRAY
X\$	VALUE TO SEARCH FOR
X1\$	ITEM TO SEARCH

FUNCTIONS USED

NAME
DIM
GOSUB
RETURN
TAB

TELEDIR

Description

This program provides you with a computerized telephone directory. It can be particularly appropriate for the names and telephone numbers of businesses or individuals that you may want to contact while you are using the computer.

Instructions for Use

Telephone numbers are added to the program as DATA statements. They must then be saved to your storage device. LOADING and RUNNING the program later will allow you to retrieve these numbers by responding to the program's messages. Retrieval is based upon the search name.

Data Entry

All data are entered using DATA statements. Each entry takes this form: name to search for, other name, and telephone number. The use of the other name item allows you to have multiple search names. See the listing for examples of data storage items.

Notes and Things to Check

The last DATA statement in your program must be STOP.

Modification Procedures

Retrieval of information is based on the first data element for each entry. If there are multiple items, all of these items will be printed along with the second (other name) item. The LEN function is used at lines 120 and 150 for formatting of output. The program continues to search for the requested name until the word "STOP" is found. The program is designed to make single searches only. If multiple searches are required, a "RESTORE" command must be added at line 335, and a "GOTO 60" must occur at line #336. It will then be necessary to use the RUN STOP key to terminate the program's operation.

```
10 REM          SAVED AT TELEDIR
20 REM THIS PROG PRODUCES A TELEPHONE
30 REM DIRECTORY COMPLETE WITH RAPID
40 REM RETRIEVAL
50 REM *****
60 PRINT CHR$(147)
70 INPUT "NAME";N0$
```

```

80 IF NO$<>" THEN 220
90 PRINT "NAME";TAB(30);"NUMBER"
100 PRINT "-----"
110 READ N$
120 L=LEN(N$)
130 IF N$="STOP" THEN 300
140 READ N1$,T$
150 L=L+LEN(N1$)
160 PRINT N$;" (";N1$;")";
170 FOR I=1 TO 24-L
180 PRINT ".";
190 NEXT I
200 PRINT T$
210 GOTO 110
220 PRINT
230 READ N$
240 IF N$="STOP" THEN GOTO 300
250 READ N1$,T$
260 IF N$<>NO$ THEN 230
270 PRINT NO$;" (";N1$;")..";T$
280 GOTO 230
290 REM *****
300 REM PROGRAM TERMINATION POINT
310 REM *****
320 PRINT
330 PRINT "END OF DATA"
340 STOP
350 REM *****
360 REM DATA ENTRY FOLLOWS
370 REM *****
380 DATA FIRE,FIRE ALARM,888-3131
390 DATA POLICE,EMERGENCY,222-3434
400 DATA KIM,STORY,602-777-1234
410 DATA JIM,BABBAGE,408-222-9876
420 DATA GEORGE,GEORGE & SUE,444-2345
430 DATA DAVE,CLEARY,513-456-1234
440 DATA JIM,JIM & CLAIR,444-9876
450 DATA STOP

```

RUN

NAME? JIM

JIM (BABBAGE)..408-222-9876

JIM (JIM & CLAIR)..444-9876

END OF DATA

Break in 340

RUN

NAME?

NAME	NUMBER
FIRE (FIRE ALARM).....	888-3131
POLICE (EMERGENCY).....	222-3434
KIM (STORY).....	602-777-1234
JIM (BABBAGE).....	408-222-9876
GEORGE (GEORGE & SUE).....	444-2345
DAVE (CLEARY).....	513-456-1234
JIM (JIM & CLAIR).....	444-9876

END OF DATA
Break in 340

MAJOR SYMBOL TABLE — TELEDIR

NAME	DESCRIPTION
I	INDEX POINTER
L	LENGTH OF DATA
N\$	NAME FROM DATA
NO\$	NAME IN
N1\$	NAME QUALIFIER (2ND ENTRY)
T\$	TELEPHONE NUMBER

FUNCTIONS USED

NAME
CHR\$
LEN

GIFTPLAN

Description

This program allows you to record and print gift lists and related items. It can be particularly useful during the Christmas period because you can estimate and evaluate costs for their effect on holiday budgets.

Instructions for Use

After you have decided on some of the gifts to be purchased, enter the information about them as DATA statements to the program. The program can then assist in projecting costs for the gifts listed for each individual. As gifts are purchased, you can enter an indicator that shows this. The resulting output provides information about total expected costs as well as the amount left to be spent.

Data Entry

All data are entered via DATA statements. Each entry contains the following information (each separated from the other by a comma): name, gift item, projected cost, and purchased indicator. A purchased indicator of Y indicates that the gift has already been bought.

Notes and Things to Check

Note that each data record must end with an N or a Y — for purchased, yes or no. If this entry is missing, the program will print error messages. Note also that the last DATA statement must contain the word STOP.

Modification Procedures

This program is a simple straight-through process with program termination always occurring at line 230. A modification you may wish to make in the future allows you to print only selected names. This can be accomplished by adding the appropriate statements in the loop at lines 80-140.

```
10 REM          SAVED AT GIFTPLAN
20 REM THIS PROGRAM PRODUCES GIFT LISTS
30 REM COSTS, AND TOTALS
40 REM *****
50 PRINT CHR$(147)
60 PRINT "NAME";TAB(15);"ITEM";TAB(30);" COST"
```

```

70 PRINT "-----"
80 READ N$
90 IF N$="STOP" THEN 160
100 READ I$,C,P$
110 T1=T1+C
120 IF P$<>"Y" THEN N=N+1:T2=T2+C:PRINT "*";
130 PRINT N$;TAB(15);I$;TAB(30);C
140 GOTO 80
150 REM *****
160 REM      PROGRAM TERMINATION POINT
170 REM *****
180 PRINT "-----"
190 PRINT "TOTAL COST";TAB(30);T1
200 PRINT "NUMBER TO BE PURCHASED";N;" COST $";T2
210 PRINT "-----"
220 PRINT
230 STOP
240 REM *****
250 REM      DATA ENTRY FOLLOWS
260 REM *****
270 DATA KIM, BARBIE, 7.95, N
280 DATA JIM, SHIRT, 14.05, Y
290 DATA GEORGE, PIPE, 2.95, Y
300 DATA JUNIOR, ELECTRIC FBALL, 19.99, Y
310 DATA STOP

```

RUN

NAME	ITEM	COST
*KIM	BARBIE	7.95
JIM	SHIRT	14.05
GEORGE	PIPE	2.95
JUNIOR	ELECTRIC FBALL	19.99
TOTAL COST		44.94
NUMBER TO BE PURCHASED 1		COST \$ 7.95

Break in 230

MAJOR SYMBOL TABLE — GIFTPLAN

NAME	DESCRIPTION
C	COST
I\$	ITEM
N\$	NAME
P\$	PURCHASED INDICATOR
T1	TOTAL COST
T2	TOTAL TO PURCHASE

FUNCTIONS USED

NAME
CHR\$
TAB

MEALPLAN

Description

This program can assist you to plan meals and to create grocery shopping lists for the menus you plan. It can be useful both in meal planning and in answering the question, "What are we having for dinner?"

Instructions for Use

RUN the program and then answer the questions the computer asks. When you are finished entering the menu for a specific meal, just press the RETURN key to move on to the next. A maximum of four meals can be entered at one time.

Data Entry

All data are entered in response to program requests.

Notes and Things to Check

To make full use of this program for planning future meals, you need a printer to provide a record of the results.

Modification Procedures

The value of N at line #60 sets the maximum number of meals that can be dealt with. It is currently set at four because of screen size limitations. If a printer is available, the number of meals can be increased and the results can be sent to the printer through a change in the PRINT statements in the printing subroutine (lines 350-540) to direct the output to the printer. Note that the three phases of the program — getting the menu items, printing the menu, and printing the grocery list — are provided in separate subroutines for ease of modification.

```
10 REM          SAVE MEALPLAN
20 REM THIS PROGRAM PROVIDES FOR MEAL
30 REM PLANNING & GROCERY LIST CREATION
40 REM *****
50 M=10
60 N=4
70 M1=100
80 DIM M$(N),I$(N,M)
90 X$="----- "
100 GOSUB 210          :REM GET DATA
110 GOSUB 350          :REM PRINT MENU
120 GOSUB 560          :REM COMPUTE GROCERIES
130 GOSUB 690          :REM PRINT LIST
```



```

140 REM *****
150 REM PROGRAM TERMINATION POINT
160 REM *****
170 PRINT
180 PRINT "PROCESSING COMPLETE"
190 STOP
200 REM *****
210 REM GET DATA
220 REM *****
230 INPUT "HOW MANY MEALS WILL YOU ENTER (1-4)";N
240 FOR I=1 TO N
250 PRINT "FOR MEAL #";I;" - INPUT DAY OF WEEK"
260 INPUT M$(I)
270 PRINT "ENTER MEAL CONTENTS-RETURN TO STOP"
280 FOR J=1 TO M
290 INPUT I$(I,J)
300 IF I$(I,J)="" THEN 320
310 NEXT J
320 NEXT I
330 RETURN
340 REM *****
350 REM PRINT DATA
360 REM *****
370 PRINT CHR$(147)
380 FOR I=1 TO N STEP 2
390 PRINT M$(I);
400 IF I+1<=N THEN PRINT TAB(20);M$(I+1)
410 IF I+1>N THEN PRINT
420 PRINT X$;
430 IF I+1<=N THEN PRINT TAB(20);X$
440 IF I+1>N THEN PRINT
450 FOR J=1 TO M
460 IF I$(I,J)="" AND I$(I+1,J)="" THEN 500
470 PRINT I$(I,J);
480 IF I+1<=N THEN PRINT TAB(20);I$(I+1,J)
490 IF I+1>N THEN PRINT
500 NEXT J
510 PRINT
520 NEXT I
530 INPUT "PRESS RETURN TO CONTINUE";A$
540 RETURN
550 REM *****
560 REM COMPUTE GROCERIES
570 REM *****
580 DIM G$(M1),G(M1)
590 FOR I=1 TO N
600 FOR J=1 TO M
610 FOR K=1 TO M1
620 IF I$(I,J)=G$(K) THEN G(K)=G(K)+1:GOTO 650
630 IF G$(K)="" THEN G$(K)=I$(I,J):M2=M2+1:G(K)=1:GOTO 650
640 NEXT K
650 NEXT J
660 NEXT I
670 RETURN
680 REM *****
690 REM PRINT LIST
700 REM *****

```

```

710 PRINT
720 PRINT TAB(10);"GROCERIES REQUIRED"
730 PRINT X$;X$
740 FOR I=1 TO M2 STEP 2
750 PRINT G(I);"..";G$(I);
760 IF I+1<=M2 THEN PRINT TAB(20);G(I+1);"..";G$(I+1);
770 PRINT
780 NEXT I
790 RETURN

```

```

RUN
HOW MANY MEALS WILL YOU ENTER (1-4)? 3
FOR MEAL # 1 - INPUT DAY OF WEEK
? SUNDAY
ENTER MEAL CONTENTS-RETURN TO STOP
? POTATOES
? ROAST BEEF
? CORN
? APPLESAUCE
? CAKE
?
FOR MEAL # 2 - INPUT DAY OF WEEK
? MONDAY
ENTER MEAL CONTENTS-RETURN TO STOP
? RICE
? PORK CHOPS
? APPLESAUCE
? GREEN BEANS
? ICE CREAM
?
FOR MEAL # 3 - INPUT DAY OF WEEK
? TUESDAY
ENTER MEAL CONTENTS-RETURN TO STOP
? POTATOES

```

```

? MEAT LOAF
? CORN
? PEACHES
?

```

SUNDAY	MONDAY
-----	-----
POTATOES	RICE
ROAST BEEF	PORK CHOPS
CORN	APPLESAUCE
APPLESAUCE	GREEN BEANS
CAKE	ICE CREAM

TUESDAY

POTATOES
MEAT LOAF
CORN
PEACHES

PRESS RETURN TO CONTINUE?

GROCERIES REQUIRED

2 ..POTATOES 1 ..ROAST BEEF
2 ..CORN 2 ..APPLESAUCE
1 ..CAKE 1 ..RICE
1 ..PORK CHOPS 1 ..GREEN BEANS
1 ..ICE CREAM 1 ..MEAT LOAF
1 ..PEACHES

PROCESSING COMPLETE
Break in 190

MAJOR SYMBOL TABLE — MEALPLAN

NAME	DESCRIPTION
G\$()	GROCERIES REQUIRED ARRAY
G()	# OF GROCERY ITEMS ARRAY
I	INDEX AND ARRAY POINTER
I\$()	MEAL ITEM ARRAY
J	INDEX AND ARRAY POINTER
K	INDEX AND ARRAY POINTER
M	MAXIMUM # OF MEAL ITEMS
M\$()	MEAL NAMES
M1	MAXIMUM NUMBER OF GROCERY ITEMS
M2	# OF ITEMS IN GROCERY ARRAY
N	MAXIMUM NUMBER OF MEALS ALLOWED
X\$	LINE OF HYPHENS

FUNCTIONS USED

NAME
CHR\$
DIM
GOSUB
RETURN

WEIGHT

Description

This program allows you to record and print the results of weight-loss efforts. It can assist you in recording successes and recognizing difficulties.

Instructions for Use

Record your weight daily, or weekly, in DATA statements. Resave the program after the data have been entered so you do not need to reenter it later. When you are ready to check your results, RUN the program and (if you are successful, you won't need to) close your eyes.

Data Entry

All data are entered via DATA statements. The form of each statement is date, weight. Note that the last entry must be DATA END.

Notes and Things to Check

To give valid comparisons, the data must be recorded on a regular basis — each day or the same day of each week. Do not include commas in the dates that you enter.

Modification Procedures

The actual acceptance of the data and the calculations are accomplished in the loop at 240-350. You may note that the function RND is used to pick a random message when you have gained weight during a period. The structure of this program is similar to that of the UTILITY program in Chapter 2, in case you wish to try some alternate processing strategies.

```
10 REM          SAVED AT WEIGHT
10 REM THIS PROGRAM RECORDS AND PRINTS
30 REM WEIGHT LOSS SUCCESSES
40 REM *****
50 M=1000
60 M$(1)="EVEN"
70 M$(2)="WHOOPS!"
80 M$(3)="A BAD WEEK"
90 M$(4)="SLIPPED"
100 M$(5)="TRY HARDER"
110 M$(6)="OH NO!"
120 M$(7)=""
```

```

130 INPUT "SHALL I PRINT A SUMMARY ONLY (Y OR N)";A$
140 PRINT
150 READ D0$,W0
160 PRINT
170 IF A$="Y" THEN 230
180 PRINT
190 PRINT "INITIAL WEIGHT WAS";W0;"ON ";D0$
200 PRINT
210 PRINT "DATE";TAB(8);"WEIGHT";TAB(16);"DIFF";TAB(22);"TOTAL"
220 PRINT "-----";TAB(8);"-----";TAB(16);"-----";TAB(22);"-----"
230 W2=W0
240 FOR I=1 TO M
250 READ D$
260 IF D$="END" THEN 360
270 READ W1
280 W3=W1-W2
290 IF W1>W2 THEN R=RND(0)*5+2
300 IF W2=W1 THEN R=1
310 W2=W1
320 IF A$="Y" THEN 340
330 PRINT D$;TAB(8);W1;TAB(16);W3;TAB(22);W2-W0;TAB(30);M$(R)
340 R=7
350 NEXT I
360 REM *****
370 REM     PROGRAM TERMINATION POINT
380 REM *****
390 IF A$="Y" THEN 440
400 PRINT TAB(22);"-----"
410 PRINT "  TOTAL LOSS IN";I-1;"PERIODS";W0-W2
420 PRINT
430 INPUT "PRESS RETURN FOR SUMMARY";A$
440 PRINT
450 PRINT "*****"
460 PRINT "FOR ";I-1;"PERIODS - STARTING ";D0$
470 PRINT "BEGINNING WEIGHT WAS:";W0
480 PRINT "CURRENT WEIGHT IS";W1
490 PRINT "TOTAL LOSS WAS";W0-W1
500 PRINT "*****"
510 PRINT
520 PRINT
530 STOP
540 REM *****
550 REM     DATA IS ENTERED HERE
560 REM *****
570 DATA JAN 1,197
580 DATA JAN 7,192
590 DATA JAN 14,193
600 DATA JAN 21,193
610 DATA JAN 28,190
620 DATA FEB 4,187
630 DATA FEB 11,188
640 DATA FEB 18,186
650 DATA FEB 25,184
660 DATA MAR 4,185
670 DATA MAR 11,182
680 DATA END

```

RUN
SHALL I PRINT A SUMMARY ONLY (Y OR N)? Y

```
*****
FOR 10 PERIODS - STARTING JAN 1
BEGINNING WEIGHT WAS: 197
CURRENT WEIGHT IS 182
TOTAL LOSS WAS 15
*****
```

Break in 530

RUN
SHALL I PRINT A SUMMARY ONLY (Y OR N)? N

INITIAL WEIGHT WAS 197 ON JAN 1

DATE	WEIGHT	DIFF	TOTAL	
JAN 7	192	-5	-5	
JAN 14	193	1	-4	SLIPPED
JAN 21	193	0	-4	EVEN
JAN 28	190	-3	-7	
FEB 4	187	-3	-10	
FEB 11	188	1	-9	WHOOFS!
FEB 18	186	-2	-11	
FEB 25	184	-2	-13	
MAR 4	185	1	-12	SLIPPED
MAR 11	182	-3	-15	

TOTAL LOSS IN 10 PERIODS 15

PRESS RETURN FOR SUMMARY?

```
*****
FOR 10 PERIODS - STARTING JAN 1
BEGINNING WEIGHT WAS: 197
CURRENT WEIGHT IS 182
TOTAL LOSS WAS 15
*****
```

Break in 530

MAJOR SYMBOL TABLE — WEIGHT

NAME	DESCRIPTION
A\$	TEMP ANSWER VARIABLE
D\$	INPUT DATE FROM DATA
D0\$	BEGIN DATE
I	INDEX AND ARRAY POINTER
M	MAXIMUM ENTRIES ALLOWED
M\$()	MESSAGE ARRAY
R	POINTER TO MESSAGE ARRAY
W0	BEGIN WEIGHT
W1	WEIGHT THIS TIME
W2	WEIGHT LAST TIME
W3	DIFFERENCE IN WEIGHT

FUNCTIONS USED

NAME
RND
TAB

4 Other Household Helpers

CHORES

Description

This program makes keeping track of household duties much simpler. It can be used to make sure that chores are shared equitably and to eliminate the "whose turn is it this week?" question.

Instructions for Use

Individual chores and persons allowed to perform them are entered as DATA statements. The program is then RUN each week, and a week number is entered from the keyboard at the computer's request. This week number is then used as the program cycles through the chores, choosing the next person on the list for the week's task. To operate effectively, the program requires that each week number be one greater than that of the previous week.

Data Entry

Data are entered in the following way:

1. The first entries of DATA statements must be the names of all individuals who will have chores.

2. Following the names, each chore must be entered and each chore entry has two parts. The first statement must begin with an asterisk, followed by its name, and end with seven entries of Y or N. Each of these Y or N entries corresponds to a day of the week and indicates whether or not that chore is performed on that day of the week. For example, an entry of

*,DISHES,Y,Y,Y,Y,Y,N,N

indicates that the chore named DISHES should be performed on days one through five but not on days six or seven. The second part of the chore

entry is a list of the individuals eligible to do that task. Note that each individual must also be named in the first DATA statements.

Notes and Things to Check

Since the DATA statements will not need to be changed often once they have been entered, data problems are unlikely after the initial runs. Note that you must keep track of each week number as you enter it and increase it by one the next week to be sure you transfer a chore to the next person in line for it.

Modification Procedures

If you have a printer available, you may want to modify the statements in lines 530-730 to provide printed output, including a checklist for task completion.

```
10 REM          SAVED AT CHORES
20 REM THIS PROGRAM MAKES KEEPING
30 REM TRACK OF HOUSEHOLD DUTIES
40 REM MUCH SIMPLER
50 REM *****
60 M=25
70 M1=10
80 DIM D0$(7),N$(M),D$(M1,7),N1$(M1),C$(M1),N0$(M)
90 FOR I=1 TO 7
100 READ D0$(I)
110 NEXT I
120 DATA S,M,T,W,T,F,S
130 FOR I=1 TO M
140 READ N0$(I)
150 IF N0$(I)="*" THEN 170
160 NEXT I
170 M=M-I-1
180 K1=K-1
190 INPUT "WEEK NUMBER";Y
200 Y=Y-1
210 GOSUB 400 :REM COMPUTE WHO
220 GOSUB 530 :REM DISPLAY RESULTS
230 REM *****
240 REM          PROGRAM TERMINATION POINT
250 REM *****
260 PRINT "PROCESSING COMPLETE"
270 PRINT
280 STOP
290 REM *****
300 REM          GET DATA
310 REM *****
320 FOR K=1 TO M
330 READ T$
340 IF T$="*" OR T$="STOP" THEN 370
```

```

350 N$(K)=T$
360 NEXT K
370 K1=K-1
380 RETURN
390 REM *****
400 REM          COMPUTE WHO
410 REM *****
420 FOR I=1 TO M
430 IF T$="STOP" THEN 500
440 T1$=T$
450 READ C$(I),D$(I,1),D$(I,2),D$(I,3),D$(I,4),D$(I,5),D$(I,6),D$(I,7)
460 GOSUB 300          :REM GET DATA
470 N=Y-K1*INT(Y/K1)+1
480 N1$(I)=N$(N)
490 NEXT I
500 S=I-1
510 RETURN
520 REM *****
530 REM          DISPLAY RESULTS
540 REM *****
550 PRINT TAB(17);"DAY OF WEEK"
560 FOR I=1 TO 7
570 PRINT TAB(10+I*4);D0$(I);
580 NEXT I
590 PRINT
600 FOR I=1 TO M
610 PRINT "-----"
620 PRINT TAB(3);N0$(I)
630 FOR J=1 TO S
640 IF N0$(I)<>N1$(J) THEN 700
650 PRINT C$(J);
660 FOR L=1 TO 7
670 PRINT TAB(10+L*4);D$(J,L);
680 NEXT L
690 PRINT
700 NEXT J
710 PRINT
720 NEXT I
730 RETURN
740 REM *****
750 REM          DATA ENTRY FOLLOWS
760 REM *****
770 DATA JAMIE
780 DATA CHUCK
790 DATA CONNIE
800 DATA KIM
810 DATA *,DISHES,Y,Y,Y,Y,Y,Y,Y
820 DATA CHUCK
830 DATA KIM
840 DATA JAMIE
850 DATA *,TRASH,Y,-,Y,-,Y,-,Y
860 DATA CONNIE
870 DATA JAMIE
880 DATA *,LAWN,-,Y,-,Y,-,Y,-
890 DATA KIM
900 DATA CHUCK
910 DATA *,VACUUM,Y,Y,Y,Y,Y,Y,Y

```

920 DATA JAMIE
930 DATA CONNIE
940 DATA STOP

RUN

WEEK NUMBER? 6

	DAY OF WEEK						
	S	M	T	W	T	F	S

JAMIE							
DISHES	Y	Y	Y	Y	Y	Y	Y
TRASH	Y	-	Y	-	Y	-	Y

CHUCK							
LAWN	-	Y	-	Y	-	Y	-

CONNIE							
VACUUM	Y	Y	Y	Y	Y	Y	Y

KIM							

PROCESSING COMPLETE

Break in 280

RUN

WEEK NUMBER? 1

	DAY OF WEEK						
	S	M	T	W	T	F	S

JAMIE							
VACUUM	Y	Y	Y	Y	Y	Y	Y

CHUCK							
DISHES	Y	Y	Y	Y	Y	Y	Y

CONNIE							
TRASH	Y	-	Y	-	Y	-	Y

KIM
LAWN - Y - Y - Y -

PROCESSING COMPLETE

Break in 280

MAJOR SYMBOL TABLE — CHORES

NAME	DESCRIPTION
C\$()	CHORE NAME ARRAY
D\$()	CHORE -DAY MATRIX ARRAY
DO\$()	DAY OF WEEK ARRAY
I	INDEX AND ARRAY POINTER
J	INDEX AND ARRAY POINTER
K	INDEX AND ARRAY POINTER
K1	COMPUTATION VARIABLE
L	INDEX AND ARRAY POINTER
M	MAXIMUM NAMES
M1	MAXIMUM CHORES
N\$()	NAMES FOR CHORES ARRAY
NO\$()	NAME ARRAY
N1\$()	NAME OF PERSON SELECTED ARRAY
S	NUMBER OF CHORES
T\$	TYPE OF RECORD INDICATOR
Y	WEEK NUMBER

FUNCTIONS USED

NAME
DIM
GOSUB
INT
RETURN
TAB

TASKS

Description

This program keeps track of required and scheduled tasks. It can be used for appointments and for projects that need to be done.

Instructions for Use

Tasks are entered and saved as DATA statements. The program is then RUN and either all items or only selected items are printed — depending on your answer to the program's questions.

Data Entry

Data are entered using DATA statements. Each entry must be provided in this form: task name, time, month, and day. If any item (such as time) is inappropriate for a task, an asterisk should be entered to indicate that this item is not used because each item must contain all four entries for the program to operate. The last entry must be END, signifying the end of data input to the program.

Notes and Things to Check

If data errors appear, then you should check to be sure that each data record has all four required entries and that each is separated from the others by a comma. Make sure that the last item is END.

Modification Procedures

Depending on the value of the month field, the print selections are either all or the M\$ array. All selections and prints-outs are performed from the data in the arrays defined in line #70. They are filled during processing by the "GET DATA" subroutine (lines 180-260).

```
10 REM          SAVED AT TASKS
20 REM  THIS PROGRAM KEEPS TRACK OF
30 REM  THINGS THAT MUST BE DONE AND
40 REM  SCHEDULED EVENTS
50 REM  *****
60 M=1000
70 DIM I$(M),T$(M),D$(M),M$(M)
80 GOSUB 180      :REM GET DATA
90 GOSUB 280      :REM PRINT RESULTS
100 REM *****
110 REM          PROGRAM TERMINATION POINT
120 REM *****
```

```

130 PRINT
140 PRINT C;"RECORDS WERE FOUND"
150 PRINT
160 STOP
170 REM *****
180 REM          GET DATA
190 REM *****
200 FOR I=1 TO M
210 READ I$(I)
220 IF I$(I)="END" THEN 250
230 READ T$(I),M$(I),D$(I)
240 NEXT I
250 M1=I-1
260 RETURN
270 REM *****
280 REM          PRINT RESULTS
290 REM *****
300 INPUT "PRINT ALL ITEMS (Y OR N)";A$
310 IF A$<>"N" THEN 330
320 INPUT "ENTER MONTH TO PRINT";M0$
330 PRINT
340 PRINT "ITEM";TAB(25);"TIME";TAB(32);"MNTH";TAB(37);"DAY"
350 PRINT "----";TAB(25);"----";TAB(32);"----";TAB(37);"----"
360 FOR I=1 TO M1
370 IF A$<>"N" THEN 390
380 IF M0$<>M$(I) THEN 410
390 PRINT I$(I);TAB(25);T$(I);TAB(32);M$(I);TAB(37);D$(I)
400 C=C+1
410 NEXT I
420 RETURN
430 REM *****
440 REM          DATA ENTRY FOLLOWS
450 REM *****
460 DATA PLANT GARDEN,* ,APR,*
470 DATA PAY TAXES,* ,APR,15
480 DATA DR. SMITH,915AM,MAY,3
490 DATA PAINT GARAGE,* ,JUL,*
500 DATA SEAN'S B-DAY,* ,JUL,30
510 DATA PAINT SHED,* ,* ,*
520 DATA END

```

```

RUN
PRINT ALL ITEMS (Y OR N)? Y

```

ITEM	TIME	MNTH	DAY
----	----	----	----
PLANT GARDEN	*	APR	*
PAY TAXES	*	APR	15
DR. SMITH	915AM	MAY	3
PAINT GARAGE	*	JUL	*
SEAN'S B-DAY	*	JUL	30
PAINT SHED	*	*	*

```

6 RECORDS WERE FOUND

```

```

Break in 160

```

```

RUN
PRINT ALL ITEMS (Y OR N)? N
ENTER MONTH TO PRINT? JUL

```

```

ITEM                TIME      MNTH DAY
-----            -
FAINT GARAGE        *        JUL  *
SEAN'S B-DAY        *        JUL 30

```

2 RECORDS WERE FOUND

Break in 160

MAJOR SYMBOL TABLE — TASKS

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
C	NUMBER OF RECORDS FOUND
D\$()	DATE ARRAY
I	INDEX AND ARRAY POINTER
I\$()	ITEM ARRAY
M	MAXIMUM RECORDS ALLOWED
M\$()	MONTH ARRAY
MO\$	MONTH TO FIND
M1	# OF RECORDS IN FILE
T\$()	TIME ARRAY

FUNCTIONS USED

NAME
DIM
GOSUB
RETURN
TAB

CALC

Description

This program causes your Commodore 64 to simulate the operation of an electronic calculator. It can be used whenever a pocket calculator is required.

Instructions for Use

Enter numbers and mathematical symbols just as you would on a calculator. After each operation, or when the equal key is pressed, the current total will be shown. If you make an error during entry, press the C. This will cause the program to clear the last entry, and to print the letters CE, for clear entry.

To return the total to zero, press the Z key. This will cause the total to return to zero and the value 000000000 to be printed.

Data Entry

Data are entered on the keyboard numeric and mathematical symbol keys.

Notes and Things to Check

The program will continue to operate until the RUN STOP key is pressed. Nonnumeric entries, except C or Z, will cause a program error.

Modification Procedures

The program receives data one character at a time from the keyboard. This information is then checked to see if it is a numeric operator. The operators (including the equal sign) indicate when the end of a number is received. The number of mathematical symbols can be increased by changing the value of the variable M and the contents of the O\$ array. Otherwise, modifications to this program are probably not desirable.

```
10 REM          SAVED AT CALC
20 REM  THIS PROGRAM SIMULATES THE USE
30 REM  OF AN ELECTRONIC CALCULATOR
40 REM  *****
50 PRINT "CALCULATOR PROGRAM"
60 PRINT "PRESS RUN/STOP TO TERMINATE"
70 M=6
80 DIM A$(10),O$(M)
```



```

90 O$(1)="*"
100 O$(2)="/"
110 O$(3)="+"
120 O$(5)="^"
130 O$(4)="-"
140 O$(6)="="
150 L$=""
160 N$=""
170 FOR I=1 TO 10
180 A$(I)=""
190 GETA$(I)
200 IF A$(I)="" THEN 190
210 IF A$(I)="C" THEN PRINT "CE":PRINT "=";T;:GOTO 150
220 IF A$(I)="-" AND I=1 AND L$<>"=" THEN 340
230 IF A$(I)<>"Z" THEN 280
240 T=0
250 PRINT
260 PRINT "000000000"
270 GOTO 150
280 FOR J=1 TO M
290 IF A$(I)<>O$(J) THEN 310
300 IF A$(I)=O$(J) THEN 370
310 NEXT J
320 IF I>1 THEN 340
330 IF J>M AND L$="" THEN PRINT:L$=""
340 N$=N$+A$(I)
350 PRINT A$(I);
360 NEXT I
370 N=VAL(N$)
380 IF N<>0 THEN 430
390 IF T=0 THEN 530
400 IF L$="" THEN 530
410 PRINT "INVALID ENTRY - MUST BE NUMERIC"
420 GOTO 160
430 IF L$<>" " THEN 460
440 T=N
450 GOTO 530
460 PRINT
470 IF L$="*" THEN T=T*N
480 IF L$="/" THEN T=T/N
490 IF L$="+" THEN T=T+N
500 IF L$="-" THEN T=T-N
510 IF L$="^" THEN T=T^N
520 PRINT "=";T;
530 L$=O$(J)
540 IF O$(J)<>"=" THEN PRINT O$(J);
550 GOTO 160
560 REM *****
570 REM NO TERMINATION - PRESS RUN/STOP
580 REM *****

```

```

RUN
CALCULATOR PROGRAM
PRESS RUN/STOP TO TERMINATE

```

```

10*10
= 100 +3
= 103 -13
= 90 /9
= 10
000000000

```

```

10+13
= 23 +3
= 26 +2
= 28 +1
= 29
100/10
= 10 ^C
Break in 200

```

MAJOR SYMBOL TABLE — CALC

NAME	DESCRIPTION
A\$()	INPUT FROM KEYBOARD
I	INDEX AND ARRAY POINTER
J	INDEX AND ARRAY POINTER
L\$	LAST SYMBOL ENCOUNTERED
M	NUMBER OF MATH SYMBOLS
N	NUMERIC NUMBER
N\$	CHARACTER VALUE OF NUMBER
O\$()	MATH SYMBOL ARRAY
T	CURRENT TOTAL

FUNCTIONS USED

NAME
DIM
GET
VAL

CAR-REC

Description

This program is designed to assist you in keeping records of a vehicle's maintenance and mileage by allowing you to record and retrieve this information. The program can be used to increase the value of your car because it keeps complete maintenance records and to improve its reliability because it encourages more consistent maintenance habits.

Instructions for Use

Enter maintenance information as DATA statements every time your vehicle is serviced or repaired. The resulting program, with its accumulated data, should then be saved on tape (or disk) for later retrieval. To print maintenance records, just enter RUN and respond to the program's request for the data that you want it to print. You can print either all data or only those records that match the value of the keyword you enter.

Data Entry

Data are entered using DATA statements. The last data item must be END to signify the end of data to the program. All data must be provided in the following form:

1. The first item must be an asterisk to indicate the beginning of the maintenance record.
2. The second item is the keyword for the maintenance type. This keyword can be used to selectively retrieve certain types of records.
3. The third item is the date. Any form is acceptable, but it must not contain a comma.
4. The next item is the mileage at the time of the maintenance.
5. The last item can be as long (as many lines) as you wish. This item can be used to record comments concerning the maintenance. Its only restriction is that it cannot contain any commas. All information entered will become a part of the record until the next line beginning with an asterisk is found.

Notes and Things to Check

If errors occur with your data, you should check to make sure that you have put commas between all the items but not included commas in your date or comments.

Modification Procedures

You can easily modify this program to record any other activities by changing the print headings and spacings in the subroutines.

```
10 REM          SAVED AT CAR-REC
20 REM THIS PROG MAINTAINS RECORDS OF
30 REM CAR MAINT AND MILES AND ALLOWS
40 REM SELECTIVE RECALL OF THOSE EVENTS
50 REM *****
60 PRINT "SHALL I PRINT ALL OF THE ITEMS (Y OR N)":INPUT A$
70 IF A$<>"Y" AND A$<>"N" THEN 60
80 IF A$="Y" THEN 100
90 INPUT "ENTER THE KEYWORD TO SEARCH FOR";X$
100 PRINT CHR$(147)
110 PRINT
120 IF A$="N" THEN GOSUB 350          :REM PRINT SELECTED ITEMS
130 IF A$="Y" THEN GOSUB 220        :REM PRINT ALL ITEMS
140 REM *****
150 REM PROGRAM TERMINATION POINT
160 REM *****
170 PRINT
180 PRINT "PROCESSING COMPLETE"
190 PRINT
200 STOP
210 REM *****
220 REM PRINT ALL ITEMS
230 REM *****
240 PRINT "KEYWORD";TAB(15);"DATE";TAB(30);"MILEAGE"
250 PRINT "-----";TAB(15);"-----";TAB(30);"-----"
260 READ T$
270 IF T$="END" THEN 330
280 IF T$<>"*" THEN PRINT T$:GOTO 260
290 PRINT
300 READ K$,D$,M
310 PRINT "      ";K$;TAB(15);D$;TAB(30);M
320 GOTO 260
330 RETURN
340 REM *****
350 REM PRINT SELECTED ITEMS
360 REM *****
370 C=0
380 PRINT "DATE";TAB(15);"MILEAGE"
390 PRINT "-----";TAB(15);"-----"
400 READ T$
410 IF T$="END" THEN 510
420 IF T$<>"*" THEN 490
430 READ K$,D$,M
440 IF X$<>K$ THEN 400
450 C=C+1
460 PRINT
470 PRINT TAB(2);D$;TAB(15);M
480 GOTO 400
490 IF X$=K$ THEN PRINT T$
```

```

500 GOTO 400
510 IF C=0 THEN PRINT "KEYWORD NOT FOUND IN DATA"
520 RETURN
530 REM *****
540 REM          DATA ENTRY FOLLOWS
550 REM *****
560 DATA *,MAINT,AUG 10 83,26543
570 DATA TUNE UP AND LUBE
580 DATA *,TIRES,SEP 5 83,28002
590 DATA REPLACED FRONT TIRES WITH BALANCE
600 DATA *,OIL,OCT 7 83,29136
610 DATA OIL CHANGE
620 DATA *,MAINT,OCT 29 83,29840
630 DATA RADIATOR FLUSH AND NEW COOLANT
640 DATA *,MILES,JAN 1 84,31674
650 DATA END

```

```

RUN
SHALL I PRINT ALL OF THE ITEMS (Y OR N)
? Y

```

KEYWORD	DATE	MILEAGE
-----	-----	-----
MAINT	AUG 10 83	26543
TUNE UP AND LUBE		
TIRES	SEP 5 83	28002
REPLACED FRONT TIRES WITH BALANCE		
OIL	OCT 7 83	29136
OIL CHANGE		
MAINT	OCT 29 83	29840
RADIATOR FLUSH AND NEW COOLANT		
MILES	JAN 1 84	31674

PROCESSING COMPLETE

Break in 200

RUN
 SHALL I PRINT ALL OF THE ITEMS (Y OR N)
 ? N
 ENTER THE KEYWORD TO SEARCH FOR? MAINT

DATE MILEAGE
 ----- -----

AUG 10 83 26543
 TUNE UP AND LUBE

OCT 29 83 29840
 RADIATOR FLUSH AND NEW COOLANT

PROCESSING COMPLETE

Break in 200

RUN
 SHALL I PRINT ALL OF THE ITEMS (Y OR N)
 ? N
 ENTER THE KEYWORD TO SEARCH FOR? NONE

DATE MILEAGE
 ----- -----

KEYWORD NOT FOUND IN DATA

PROCESSING COMPLETE

Break in 200

MAJOR SYMBOL TABLE — CAR-REC

NAME	DESCRIPTION
C	COUNT OF MATCHING RECORDS
D\$	DATE
K\$	KEYWORD
M	MILEAGE
T\$	RECORD TYPE/COMMENTS

FUNCTIONS USED

NAME
CHR\$
GOSUB
RETURN
TAB

JOB-EST

Description

This program is designed to assist you in estimating costs and materials for home (or other) maintenance or construction tasks. It can be used in a wide variety of home activities that require the estimation of costs when multiple quantities of several items may be required.

Instructions for Use

Enter the items that will be required, including their costs, in DATA statements. Then RUN the program and review the results. You may want to add, delete, or change some of the items previously entered. The program will lead you through a sequence that permits you to make these changes. To retain the data for future use, you should save the program on tape (or disk).

Data Entry

Data are entered using DATA statements as is illustrated in lines 910-990 of the sample listing. Note that line 1000 ends the data and that the last data item must be END. Each data item contains item name, quantity, and cost.

Notes and Things to Check

If the program operates incorrectly, you should check to make sure that the data items are separated by commas (and that you did not mistakenly press the period to separate the entries or the comma instead of the decimal point for the cost field). If the data are misaligned because of this, the program will reject the item name as erroneous if it occurs when the program is expecting quantity or cost. This type of error will normally not be seen until a line after the offending entry.

Modification Procedures

This program has been provided in the form of three independent subroutines to allow ease of modification in any of the areas. For example, to change the input source to tape (or disk) instead of DATA statements, you need to change only the instructions in the GET INPUT subroutine at line 190. This will not interfere with the other program processes.

```

10 REM          SAVED AT JOB-EST
20 REM  THIS PROGRAM ESTIMATES THE COST
30 REM  COMPLETING A HOME MAINTENANCE
40 REM  OR CONSTRUCTION TASK
50 REM  *****
60 M=100
70 DIM N$(M),C(M),Q(M)
80 GOSUB 190      :REM  GET INPUT
90 GOSUB 300      :REM  CHANGE INPUT
100 GOSUB 710     :REM  PRINT RESULTS
110 REM  *****
120 REM  PROGRAM TERMINATION POINT
130 REM  *****
140 PRINT
150 PRINT "PROCESSING COMPLETE"
160 PRINT
170 STOP
180 REM  *****
190 REM          GET INPUT
200 REM  *****
210 READ P$
220 FOR I=1 TO M
230 READ N$(I)
240 IF N$(I)="END" THEN 270
250 READ Q(I),C(I)
260 NEXT I
270 M1=I-1
280 RETURN
290 REM  *****
300 REM          CHANGE INPUT
310 REM  *****
320 A$=""
330 PRINT "WOULD YOU LIKE TO CHANGE ANY ITEMS (Y OR N)":INPUT A$
340 IF A$<>"Y" THEN 690
350 A1$=""
360 INPUT "ADD(A), DELETE(D), OR CHANGE(C)";A1$
370 IF A1$="D" THEN 410
380 IF A1$="A" THEN 510
390 IF A1$="C" THEN 570
400 GOTO 330
410 PRINT "***** DEL ITEM *****"
420 INPUT "ITEM DESC";N0$
430 FOR I=1 TO M1
440 IF N$(I)<>N0$ THEN 470
450 N$(I)=" "
460 GOTO 350
470 NEXT I
480 PRINT "ITEM NOT FOUND"
490 GOTO 350
500 IF A1$="C" THEN 570
510 PRINT "***** ADD ITEM *****"
520 M1=M1+1
530 INPUT "ITEM DESC";N$(M1)
540 INPUT "QTY";Q(M1)
550 INPUT "COST";C(M1)
560 GOTO 350
570 PRINT "***** CHANGE ITEM *****"

```



```

580 INPUT "ITEM DESC TO CHANGE";N0$
590 FOR I=1 TO M1
600 IF N$(I)<>N0$ THEN 660
610 INPUT "CHANGE DESC(D), QTY(Q), OR COST(C)";A2$
620 IF A2$="D" THEN INPUT "NEW DESC";N$(I)
630 IF A2$="Q" THEN INPUT "NEW QTY";Q(I)
640 IF A2$="C" THEN INPUT "NEW COST";C(I)
650 GOTO 350
660 NEXT I
670 PRINT "ITEM NOT FOUND"
680 GOTO 350
690 RETURN
700 REM *****
710 REM          PRINT RESULTS
720 REM *****
730 PRINT CHR$(147)
740 PRINT
750 PRINT P$
760 PRINT
770 PRINT "MATERIAL";TAB(18);"QTY";TAB(24);"COST";TAB(32);"TOT COST"
780 PRINT "-----";TAB(18);"---";TAB(24);"-----";TAB(32);"-----"
790 FOR I=1 TO M1
800 IF N$(I)=" " THEN 840
810 T0=Q(I)*C(I)
820 PRINT N$(I);TAB(17+Q1);Q(I);TAB(23+C1);C(I);TAB(31+T1);T0
830 T=T+T0
840 NEXT I
850 PRINT "-----"
860 PRINT TAB(20);"TOTAL COST ";T
870 RETURN
880 REM *****
890 REM          DATA ENTRY FOLLOWS
900 REM *****
910 DATA BRICK BBQ
920 DATA NEW USED BRICKS,405,.26
930 DATA FIREBRICK,40,.45
940 DATA MORTAR MIX,6,5.95
950 DATA SAND,3,45.90
960 DATA FLAGSTONE,400,.1
970 DATA 18X24 DOORS,2,54.94
980 DATA GRILL,1,129.95
990 DATA ROTISSERIE,1,34.83
1000 DATA END

```

RUN
WOULD YOU LIKE TO CHANGE ANY ITEMS (Y OR N)
? N

BRICK BBQ

MATERIAL	QTY	COST	TOT COST
NEW USED BRICKS	405	.26	105.3
FIREBRICK	40	.45	18
MORTAR MIX	6	5.95	35.7
SAND	3	45.9	137.7
FLAGSTONE	400	.1	40
18X24 DOORS	2	54.94	109.88
GRILL	1	129.95	129.95
ROTISSERIE	1	34.83	34.83
TOTAL COST			611.36

PROCESSING COMPLETE

Break in 170

RUN
WOULD YOU LIKE TO CHANGE ANY ITEMS (Y OR N)
? Y
ADD(A), DELETE(D), OR CHANGE(C)? C
***** CHANGE ITEM *****
ITEM DESC TO CHANGE? SAND
CHANGE DESC(D), QTY(Q), OR COST(C)? D
NEW DESC? MORTAR SAND
ADD(A), DELETE(D), OR CHANGE(C)? A
***** ADD ITEM *****
ITEM DESC? NEW ITEM
QTY? 1
COST? 5.55
ADD(A), DELETE(D), OR CHANGE(C)? D
***** DEL ITEM *****
ITEM DESC? GRILL
ADD(A), DELETE(D), OR CHANGE(C)?
WOULD YOU LIKE TO CHANGE ANY ITEMS (Y OR N)
?N

BRICK BBQ

MATERIAL	QTY	COST	TOT COST
NEW USED BRICKS	405	.26	105.3
FIREBRICK	40	.45	18
MORTAR MIX	6	5.95	35.7
MORTAR SAND	3	45.9	137.7
FLAGSTONE	400	.1	40
18X24 DOORS	2	54.94	109.88
ROTISSERIE	1	34.83	34.83
NEW ITEM	1	1.55	1.55

TOTAL COST 482.96

PROCESSING COMPLETE

Break in 170

MAJOR SYMBOL TABLE — JOB-EST

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
A1\$	ANSWER VARIABLE
A2\$	ANSWER VARIABLE
C()	ITEM COST ARRAY
I	INDEX AND ARRAY POINTER
J	INDEX AND ARRAY POINTER
M	MAXIMUM NUMBER OF ENTRIES
M1	ACTUAL NUMBER OF ITEMS
N\$()	ITEM NAME ARRAY
P\$	PROJECT NAME
Q()	QUANTITY ARRAY

FUNCTIONS USED

NAME
CHR\$
DIM
GOSUB
RETURN
TAB

5 For the Home Worker

WORK-REC

Description

This program records and calculates work and vacation hours. It can be used to record and/or project vacation usage and/or availability.

Instructions for Use

At the end of each pay period, you should record hours worked, holiday hours, vacation hours, and sick time (in hours). This information should then be saved to tape (or disk). The program can be run at any time to produce work/vacation hour records. Projected data can also be entered (but not saved) to check projections of future vacation plans.

Data Entry

All data are entered as DATA statements. Two separate statements are required to initialize the date before you can enter weekly records. The first entry contains the following information: hours in pay period, vacation hours earned in the pay period, and sick hours earned during the pay period. The second DATA statement contains the number of vacation hours available at the start of the program's usage, followed by the number of hours for sick time.

Following these initialization entries, each pay period can be entered on a single line in the following form: date, hours worked, holiday hours, vacation hours used, and sick hours used. Make sure that each of these items is separated from the others by a comma and that no comma appears elsewhere in the entry.

Notes and Things to Check

For this program, an asterisk data entry as the date will terminate the program's search for data. This item should follow all other DATA

statements. Because this program inputs all data before it prints any output, an error in the DATA statements will occur before any noticeable program activity takes place. If an error message points to a line number less than 200, or to a DATA statement, you should look at the data for the incorrect alignment of information.

Modification Procedures

All data are entered to arrays before the results are printed. If you want to modify the calculation procedures, you can do so at lines 160 and 170 without affecting the print activity.

```
10 REM      SAVED AT WORK-REC
20 REM      THIS PROGRAM RECORDS WORK AND
30 REM      VACATION HOURS
40 REM      *****
50 M=52
60 DIM W(M),H(M),V(M),S(M)
70 READ A0,V0,S0
80 READ V1,S1
90 T3=V1
100 T4=S1
110 FOR I=1 TO M
120 READ D$
130 IF D$="*" THEN 190
140 D9$=D$
150 READ W(I),H(I),V(I),S(I)
160 T3=T3+V0
170 T4=T4+S0
180 NEXT I
190 M1=I-1
200 GOSUB 290      :REM PRINT RESULTS
210 REM      *****
220 REM      PROGRAM TERMINATION POINT
230 REM      *****
240 PRINT
250 PRINT "PROCESSING COMPLETE"
260 PRINT
270 STOP
280 REM      *****
290 REM      PRINT RESULTS
300 REM      *****
310 W9=W(M1)+H(M1)+V(M1)+S(M1)
320 IF W9 =A0 THEN 450
330 PRINT
340 PRINT "ERRONEOUS HOUR ENTRY - TOTAL INCORRECT"
350 PRINT
360 PRINT "TOTAL HOURS AVAILABLE";A0
370 PRINT "WORK ";TAB(15);W(M1)
380 PRINT "HOLIDAY";TAB(15);H(M1)
```

```

390 PRINT "VACATION";TAB(15);V(M1)
400 PRINT "SICK TIME";TAB(15);S(M1)
410 PRINT TAB(15);"-----"
420 PRINT "TOTAL ";TAB(22);W9
430 PRINT "DIFFERENCE";TAB(22);A0-W9
440 GOTO 580
450 PRINT
460 PRINT "FOR:";D9$
470 PRINT
480 PRINT TAB(10);"WORK";TAB(15);"HOL";TAB(22);"VAC";
490 PRINT TAB(29);"SICK";TAB(36);"TOT"
500 PRINT TAB(10);"HRS ";TAB(15);"HRS";TAB(22);"HRS";
510 PRINT TAB(29);"HRS ";TAB(36);"HRS"
520 PRINT "-----"
530 PRINT "AVAIL";TAB(22);T3;TAB(28);T4;TAB(35);T3+T4
540 PRINT "THIS WK";TAB(10);W(M1);TAB(15);H(M1);TAB(22);V(M1);
550 PRINT TAB(28);S(M1);TAB(35);V(M1)+S(M1)
560 PRINT TAB(22);"-----"
570 PRINT "CUM";TAB(22);T3-V(M1);TAB(28);T4-S(M1);TAB(35);T3+T4-V(M1)-
S(M1)
580 RETURN
590 REM *****
600 REM DATA AREA
610 REM *****
620 DATA 40,4,2
630 DATA 16,8
640 DATA WEEK1,29,8,2,1
650 DATA WEEK2,40,0,0,0
660 DATA WEEK3,32,0,5,3
670 DATA "*"

```

RUN

FOR:WEEK3

	WORK HRS	HOL HRS	VAC HRS	SICK HRS	TOT HRS
AVAIL			28	14	42
THIS WK	32	0	5	3	8
CUM			23	11	34

PROCESSING COMPLETE

Break in 270

MAJOR SYMBOL TABLE — WORK-REC

NAME	DESCRIPTION
A0	HOURS IN NORMAL WEEK
D\$	DATE OF CURRENT WEEK
D9\$	DATE OF LAST WEEK
H()	HOLIDAY HOUR ARRAY
I	INDEX AND ARRAY POINTER
M	MAXIMUM NUMBER OF ENTRIES
M1	ACTUAL NUMBER OF ENTRIES
S()	SICK TIME ARRAY
S0	SICK TIME ACCRUED IN WEEK
S1	INITIAL SICK TIME AVAILABLE
T3	TOTAL VACATION
T4	TOTAL SICK TIME AVAIL
V()	VACATION HOUR ARRAY
V0	VACATION TIME ACCRUED IN WEEK
V1	INITIAL VACATION AVAILABLE
W\$	STRING OF W
W()	WORK HOURS IN WEEK
W9	HOURS ACCOUNTED FOR

FUNCTIONS USED

NAME
DIM
GOSUB
RETURN
TAB

PROFLOSS

Description

This program produces a current and cumulative profit and loss picture for your home business activity. It can be used in a wide variety of home businesses from raising and selling pets to running a catalog sales operation.

Instructions for Use

At the end of each business period, you should enter your income and expense information as DATA statements in the program. When you RUN the program, either at that time or later, you will see a quick synopsis of your business profit or loss for the most recent period, and cumulative profit or loss since you started to use the program. Note that the last data item must be an asterisk. In addition, you should always resave your program to tape or disk after each entry of new data.

Data Entry

This program receives its information from DATA statements. The first statement provides the income that you want the program to begin with, followed by the expenses you want the program to aggregate with the other data you will be entering later. This entry allows you to tune the program to your business accounting, or activity, period.

The remaining entries should be made as each business period ends. Each of these entries is of this form: date, income, and expenses. One entry is made for each business period. The last entry is a DATA statement containing only an asterisk to indicate the end of data to the computer.

Notes and Things to Check

Remember to separate each entry with a comma and use no commas elsewhere in the data. Also remember that an asterisk tells the computer it has received the last data item, so always be sure that it is used nowhere else.

Modification Procedures

All data are entered into arrays (E and I). Additional data can be entered if you add arrays and change the read statement that occurs at line 130 of the program. These additional data items can then be manipulated and printed out through the modification of the print formats in the

output subroutine. Note that M (at line 40) sets the maximum number of entries that can be recorded in the program.

```

10 REM      SAVED AT PROFLOSS
20 REM THIS PROGRAM RECORDS PROFIT AND
25 REM LOSS FROM A BUSINESS ACTIVITY
30 X9$="*****"
40 M=52
50 DIM E(M),I(M)
60 READ I1,E1
70 T1=I1
80 T2=E1
90 FOR J=1 TO M
100 READ D$
110 IF D$="*" THEN 170
120 D9$=D$
130 READ I(J),E(J)
140 T1=T1+I(J)
150 T2=T2+E(J)
160 NEXT J
170 M1=J-1
180 GOSUB 270      :REM PRINT RESULTS
190 REM *****
200 REM      PROGRAM TERMINATION POINT
210 REM *****
220 PRINT
230 PRINT "PROCESSING COMPLETE"
240 PRINT
250 STOP
260 REM *****
270 REM      PRINT RESULTS
280 REM *****
290 T1=T1-I(M1)
300 T2=T2-E(M1)
310 PRINT
320 PRINT X9$
330 PRINT "AS OF: ";D9$
340 PRINT
350 PRINT TAB(12);"INCOME";TAB(21);"EXPENSE";TAB(32);"NET"
360 PRINT TAB(14);" $ ";TAB(23);" $ ";TAB(32);" $ "
370 PRINT "-----"
380 PRINT "PAST ";TAB(12);T1;TAB(21);T2;TAB(30);T1-T2
390 PRINT "THIS PERIOD";TAB(12);I(M1);
400 PRINT TAB(21);E(M1);TAB(30);I(M1)-E(M1)
410 PRINT TAB(12);"-----"
420 PRINT "TOTAL";TAB(12);T1+I(M1);TAB(21);T2+E(M1);TAB(30);T1-T2+I(M1)
    )-E(M1)
430 PRINT X9$
440 RETURN
450 REM *****
460 REM      DATA AREA
470 REM *****

```

```

480 DATA 1000,500
490 DATA WEEK1,1000,500
500 DATA WEEK2,1000,500
510 DATA WEEK3,1000,500
520 DATA "*"

```

RUN

AS OF:WEEK3

	INCOME	EXPENSE	NET
	\$	\$	\$
PAST	3000	1500	1500
THIS PERIOD	1000	500	500
TOTAL	4000	2000	2000

PROCESSING COMPLETE

Break in 250

MAJOR SYMBOL TABLE — PROFLOSS

NAME	DESCRIPTION
D\$	DATE
D9\$	LAST DATE
E()	EXPENSE ARRAY
E1	INITIAL EXPENSES
I()	INCOME ARRAY
I1	INITIAL INCOME
J	INDEX AND ARRAY POINTER
M	MAXIMUM ENTRIES ALLOWED
M1	NUMBER OF ENTRIES
T1	TOTAL INCOME
T2	TOTAL EXPENSES
X9\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
DIM
GOSUB
TAB

ORDER-FM

Description

This program provides an automated order entry system for home use. It can be used to assist the home-worker who is involved in any type of sales activity.

Instructions for Use

Before using the program for the first time, you must enter each product you are selling as a DATA statement to the program, including pertinent information such as unit of sale, price, and other items. Any time after these data have been entered, you RUN the program and fill in orders through the keyboard in response to the program's messages. Orders will continue to be accepted until you respond to the "CUSTOMER NAME?" question with the word STOP. Unless you enter STOP, the program will continue to accept and process new orders, printing each as it occurs and storing the result to a tape (or disk) file.

Data Entry

Products are entered as DATA statements, in the form of product number, product name, unit of sale, and price. Note that product number is the key used for order-entry procedures, so this must be unique for each product. Individual orders against these items are entered via the keyboard in response to the program's requests. Products will continue to be added to an order until a RETURN is found without any data, and new customer orders will continue to be requested until a STOP is found for the customer's name. These responses terminate their respective activities.

Notes and Things to Check

This program requires a tape or disk for storage. As a result, if you terminate the program in an abnormal way (such as pressing the RUN STOP key), you may lose data. You can minimize these problems by carefully following the instructions in your Commodore User's Manual for the storage device you are using. As a general rule, you should avoid ending a processing session without using the normal program processes.

Note: This program operates in conjunction with the ORDER-PR program, which allows you to retrieve the orders this program writes to storage.

Modification Procedures

Remember that this program is normally meant to operate in conjunction with ORDER-PR if you attempt to modify or customize it. In general, you should find that both programs can easily be modified through changes to the subroutines that perform specific tasks. For example, you can locate product files on disk or tape and then retrieve them for the program's use by adding an OPEN statement and then changing the READ statements to INPUT#, in the "Get Products" subroutine.

```
10 REM SAVED AT ORDER-FM
20 REM THIS PROGRAM PRODUCES AN ORDER
30 REM FORM FOR HOME SALES USE
40 REM *****
50 R9$=CHR$(13)
60 M2=50
70 M=100
80 DIM N$(M),P$(M),U$(M),P(M),Q0(M2),N0(M2)
90 INPUT "TODAY'S DATE:";D$
100 GOSUB 1130 :REM OPEN FILES
110 GOSUB 270 :REM GET PRODUCTS
120 GOSUB 430 :REM ACCEPT ORDER
130 IF C1$="STOP" THEN 180
140 GOSUB 720 :REM PRINT ORDER
150 GOSUB 930 :REM STORE ORDER
160 GOTO 120
170 REM *****
180 REM PROGRAM TERMINATION POINT
190 REM *****
200 CLOSE 1
210 PRINT
220 PRINT
230 PRINT "PROCESSING COMPLETE"
240 PRINT
250 STOP
260 REM *****
270 REM GET PRODUCTS
280 REM *****
290 DATA 1,PRODUCT 1,EA,1.45
300 DATA 2,PRODUCT 2,QT,1.22
310 DATA 3,PRODUCT 3,BX,8.52
320 DATA 4,PRODUCT 4,EA,.32
330 DATA *
340 I=1
350 READ N$(I)
360 IF N$(I)="*" THEN 400
370 READ P$(I),U$(I),P(I)
380 I=I+1
390 GOTO 350
400 M=I-1
410 RETURN
```

```

420 REM *****
430 REM          ACCEPT ORDER
440 REM *****
450 PRINT CHR$(147)
460 J=0
470 M3=0
480 C1$=""
490 INPUT "CUSTOMER NAME: ";C1$
500 IF C1$="" THEN 490
510 IF C1$="STOP" THEN 700
520 INPUT "CUSTOMER ADDRESS: ";C2$
530 INPUT "          :";C3$
540 INPUT "TELEPHONE: ";T$
550 PRINT
560 REM **** ADD PRODUCTS *****
570 J=J+1
580 N0$=""
590 INPUT "PRODUCT #";N0$
600 IF N0$="" THEN 690
610 GOSUB 1020 :REM FIND PRODUCT
620 IF P0=0 THEN 670
630 N0(J)=P0
640 PRINT N$(P0);TAB(8);P$(P0);TAB(25);U$(P0);TAB(29);P(P0)
650 INPUT "QUANTITY";Q0(J)
660 IF Q0(J)<>0 THEN 570
670 PRINT "ENTRY IGNORED"
680 GOTO 580
690 M3=J-1
700 RETURN
710 REM *****
720 REM          PRINT ORDER
730 REM *****
740 PRINT CHR$(147)
750 T=0
760 PRINT C1$
770 PRINT C2$
780 PRINT C3$;TAB(30);T$
790 PRINT
800 PRINT "# PROD#";TAB(10);"PRODUCT";TAB(25);"UNIT";TAB(29);"$";TAB(35);"QTY"
810 FOR I=1 TO M3
820 PRINT I;N$(N0(I));TAB(10);P$(N0(I));TAB(25);U$(N0(I));TAB(29);P(N0(I));
830 PRINT TAB(35);Q0(I)
840 T=T+P(N0(I))*Q0(I)
850 NEXT I
860 PRINT TAB(10);"TOTAL PRICE ";TAB(27);"$";T
870 INPUT "CORRECTIONS (Y OR N)";A$
880 IF A$<>"Y" THEN 910
890 GOSUB 1180 :REM CORRECTIONS
900 GOTO 740
910 RETURN
920 REM *****
930 REM          STORE ORDER
940 REM *****
950 PRINT#1,C1$;R9$;C2$;R9$;C3$;R9$;T$
960 FOR I=1 TO M3

```

```

970 PRINT#1,N$(NO(I));R9$;P$(NO(I));R9$;P(NO(I));R9$;Q0(I)
980 NEXT I
990 PRINT#1,"*"
1000 RETURN
1010 REM *****
1020 REM          FIND PRODUCT
1030 REM *****
1040 P0=0
1050 FOR I=1 TO M
1060 IF N0$=N$(I) THEN 1100
1070 NEXT I
1080 PRINT "PRODUCT # NUMBER NOT FOUND"
1090 GOTO 1110
1100 P0=I
1110 RETURN
1120 REM *****
1130 REM          OPEN FILES
1140 REM *****
1150 OPEN 1,1,1,D$
1160 RETURN
1170 REM *****
1180 REM          CORRECTIONS
1190 REM *****
1200 INPUT "ENTER # FROM ORDER";I
1210 IF I=0 THEN 1260
1220 IF I<=M3 THEN INPUT "ENTER NEW QTY";Q0(I):GOTO 1200
1230 PRINT "ADD PRODUCTS"
1240 J=M3
1250 GOSUB 560          :REM ADD PRODUCTS
1260 RETURN

```

RUN

TODAY'S DATE: ? JAN 1 84

CUSTOMER NAME: ? JOE E DOE
CUSTOMER ADDRESS: ? 455 SUMMIT RIDGE
: ? TRUCKEE CA 12345
TELEPHONE: ? 123-456-7890

PRODUCT # ? 4
4 PRODUCT 4 EA .32
QUANTITY ? 3
PRODUCT # ? 1
1 PRODUCT 1 EA 1.45
QUANTITY ? 2
PRODUCT # ?

JOE E DOE
455 SUMMIT RIDGE
TRUCKEE CA 12345
123-456-7890

#	PROD#	PRODUCT	UNIT\$	QTY
1	4	PRODUCT 4	EA .32	3
2	1	PRODUCT 1	EA 1.45	2
		TOTAL PRICE	\$ 3.86	

CORRECTIONS (Y OR N)? N

CUSTOMER NAME:? JAMES E SMITH
CUSTOMER ADDRESS:? 123 WINTON ROAD
 :? DALLAS TX 98765
TELEPHONE:? 567-4321

PRODUCT #? 3
3 PRODUCT 3 BX 8.52
QUANTITY? 1
PRODUCT #? 4
4 PRODUCT 4 EA .32
QUANTITY? 2
PRODUCT #?

JAMES E SMITH
123 WINTON ROAD
DALLAS TX 98765 567-4321

#	PROD#	PRODUCT	UNIT\$	QTY
1	3	PRODUCT 3	BX 8.52	1
2	4	PRODUCT 4	EA .32	2
		TOTAL PRICE	\$ 9.16	

CORRECTIONS (Y OR N)? N

CUSTOMER NAME:? STOP

PROCESSING COMPLETE

Break in 250

RUN
TODAY'S DATE:? JAN 2 84

CUSTOMER NAME:? JANE DOAKS
CUSTOMER ADDRESS:? 765 RIDER STREET
 :? LOS ANGELES CA 12345

TELEPHONE:? 657-543-1234

PRODUCT #? 2
2 PRODUCT 2 QT 1.22
QUANTITY? 1

PRODUCT #? 1
1 PRODUCT 1 EA 1.45
QUANTITY? 2
PRODUCT #? 4
4 PRODUCT 4 EA .32
QUANTITY? 2
PRODUCT #?

JANE DOAKS
765 RIDER STREET
LOS ANGELES CA 12345
657-543-1234

#	PROD#	PRODUCT	UNIT\$	QTY
1	2	PRODUCT 2	QT 1.22	1
2	1	PRODUCT 1	EA 1.45	2
3	4	PRODUCT 4	EA .32	2
TOTAL PRICE			\$ 4.76	

CORRECTIONS (Y OR N)? Y

ENTER # FROM ORDER? 5

ADD PRODUCTS

PRODUCT #? 4

4 PRODUCT 4 EA .32

QUANTITY? 5

PRODUCT #?

JANE DOAKS
765 RIDER STREET
LOS ANGELES CA 12345
657-543-1234

#	PROD#	PRODUCT	UNIT\$	QTY
1	2	PRODUCT 2	QT 1.22	1
2	1	PRODUCT 1	EA 1.45	2
3	4	PRODUCT 4	EA .32	2
4	4	PRODUCT 4	EA .32	5
TOTAL PRICE			\$ 6.36	

CORRECTIONS (Y OR N)? N

CUSTOMER NAME:? STOP

PROCESSING COMPLETE

Break in 250

MAJOR SYMBOL TABLE — ORDER-FM

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
C1\$	CUSTOMER NAME
C2\$	CUSTOMER ADDRESS LINE 1
C3\$	CUSTOMER ADDRESS LINE 2
D\$	DATE
I	INDEX AND ARRAY POINTER
J	INDEX AND ARRAY POINTER
M	MAXIMUM NUMBER OF PRODUCTS
M1	MAXIMUM NUMBER OF CUSTOMERS
M2	MAXIMUM LINES TO PRINT
M3	NUMBER OF PRODUCTS ENTERED
N\$()	PRODUCT NUMBER ARRAY
NO\$	PRODUCT NUMBER TO ADD
NO()	POINTER ARRAY TO SELECTED ITEMS
P\$()	PRODUCT NAME ARRAY
P()	PRODUCT PRICE ARRAY
P0	POINTER TO PRODUCT FOUND
Q0()	QUANTITY ARRAY
R9\$	CARRIAGE RETURN
T	TOTAL PRICE
T\$	CUSTOMER TELEPHONE NUMBER
U\$()	PRODUCT UNIT OF SALE ARRAY

FUNCTIONS USED

NAME
CHR\$
CLOSE
DIM
GOSUB
OPEN
RETURN
TAB

ORDER-PR

Description

This program works in conjunction with the previous program, ORDER-FM, to print out orders received and recorded by that program.

Instructions for Use

RUN the program and enter the date on which the file was created by ORDER-FM. This date is the file name under which the information is stored.

Data Entry

All data are provided by the program ORDER-FM.

Notes and Things to Check

If the date given to this program is not exactly the same as the date provided to ORDER-FM, your computer will be unable to find the file for processing. If this happens, you must establish a consistent way of entering dates. In any case, you will find it desirable in your use of these programs to provide adequate records of their storage locations and file names for your orders.

Modification Procedures

Because this program operates on data provided by ORDER-FM, you have little flexibility in changing it. The output subroutine can be altered since it is independent of the data input function. For example, you may want to provide printer output by changing the print formats in this subroutine.

```
10 REM      SAVED AT ORDER-PR
20 REM      PRINT OF ORDER RECORDS FROM
30 REM FILES CREATED BY ORDER-FM
40 X$="*****"
50 M=20
60 DIM N$(M),P$(M),P(M),Q0(M)
70 PRINT "ENTER THE DATE OF THE INPUT FILE";
80 F1$=""
90 INPUT F1$
100 IF F1$="" THEN 180
110 OPEN 1,1,0,F1$
120 GOSUB 260          'PRINT DATA
130 I1=I
```

```

140 PRINT I1;"RECORDS WERE FOUND"
150 CLOSE 1
160 GOTO 70
170 REM *****
180 REM   PROGRAM TERMINATION POINT
190 REM *****
200 PRINT
210 PRINT
220 PRINT "PROCESSING COMPLETE"
230 PRINT
240 STOP
250 REM *****
260 REM           PRINT DATA
270 REM *****
280 FOR I=1 TO M
290 GOSUB 450 :REM GET INPUT
300 PRINT
310 PRINT "#";I;
320 PRINT "CUSTOMER:";TAB(15);C1$;"   ";T$
330 PRINT TAB(15);C2$
340 PRINT TAB(15);C3$
350 PRINT "ORDER:"
360 FOR J=1 TO M4
370 PRINT N$(J);TAB(10);P$(J);TAB(29);P(J);TAB(35);Q0(J)
380 NEXT J
390 PRINT
400 IF SR<>0 THEN 420
410 NEXT I
420 PRINT "EOF"
430 RETURN
440 REM *****
450 REM           GET INPUT
460 REM *****
470 M4=0
480 INPUT#1,C1$,C2$,C3$,T$
490 SR=ST:IF SR<>0 THEN 570
500 FOR J=1 TO M
510 INPUT#1,N$(J)
520 IF N$(J)="*" THEN 550
530 INPUT#1,P$(J),P(J),Q0(J)
540 NEXT J
550 M4=J-1
555 SR=ST
560 SR=ST
570 RETURN

```

RUN

ENTER THE DATE OF THE INPUT FILE? JAN 2
B4

1 CUSTOMER: JANE DOAKS 657-543-1234
765 RIDER STREET
LOS ANGELES CA 12345

ORDER:

2 PRODUCT 2 1.22 1

```

1      PRODUCT 1      1.45  2
4      PRODUCT 4      .32   2
4      PRODUCT 4      .32   5

```

EOF

```

1 RECORDS WERE FOUND
ENTER THE DATE OF THE INPUT FILE?

```

PROCESSING COMPLETE

Break in 240

MAJOR SYMBOL TABLE — ORDER-PR

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
C1\$	CUSTOMER NAME
C2\$	CUSTOMER ADDRESS LINE 1
C3\$	CUSTOMER ADDRESS LINE 2
D\$	DATE
I	INDEX AND ARRAY POINTER
I1	NUMBER OF ORDERS IN FILE
J	INDEX AND ARRAY POINTER
M	MAXIMUM NUMBER OF PRODUCTS
M1	MAXIMUM NUMBER OF CUSTOMERS
M2	MAXIMUM LINES TO PRINT
M3	NUMBER OF PRODUCTS ENTERED
M4	NUMBER OF LINES IN ORDER
N\$()	PRODUCT NUMBER ARRAY
N0\$	PRODUCT NUMBER TO ADD
N0()	POINTER ARRAY TO SELECTED ITEMS
P\$()	PRODUCT NAME ARRAY
P()	PRODUCT PRICE ARRAY
P0	POINTER TO PRODUCT FOUND
Q0()	QUANTITY ARRAY
SR	SAVED I/O STATUS
T	TOTAL PRICE
T\$	CUSTOMER TELEPHONE NUMBER
U\$()	PRODUCT UNIT OF SALE ARRAY

FUNCTIONS USED

NAME
CHR\$
CLOSE
DIM
GOSUB
OPEN
RETURN
ST
TAB

BIDCREAT

Description

This program is designed to assist the home business person in the tedious task of estimating itemized costs and prices. It can be used in any activity that requires such estimates.

Instructions for Use

Before using this program, you must enter an itemized list of your standard bid items as DATA statements in the program. The data in the sample listing are shown in lines 340-380. Once you have entered these data records, you can RUN the program at any time to produce the desired results. The "BID NAME?" entry requested by the computer is used as the name of the file for storage to tape or disk.

Data Entry

Standard items must be entered as DATA statements to the program. These items are required to be in the following form:

PRODUCT NUMBER, NAME, UNIT TO BID, COST, PRICE

The last DATA statement must contain only an asterisk, which signifies the end of data to the program. Additional items (nonstandard items) can be added to the bid during the program's operation, but unlike those that appear in the DATA statements, they will not remain available for later use.

Notes and Things to Check

You should insure that all of the five required entries are provided for each record in your data and that a comma separates each entry. Your output may show costs, prices, or totals with a large number of decimal positions, which are generated by the Commodore's internal format for storing the numeric items (as approximations). Although these numbers may appear strange, you should find that they are very close to the expected results.

Modification Procedures

This program has been completely separated into subroutines for ease of modification. If you have become familiar with the ORDER-FM program shown earlier, you will notice marked similarities in the two programs' structures. An understanding of one program will help you

with the other, but there are substantial differences between them in their methods of internal operation. This is particularly true of their methods of data input.

```
10 REM          SAVED AT BIDCREAT
20 REM  THIS PROGRAM ACCEPTS INPUTS AND
30 REM  PRODUCES INFORMATION FOR BIDDING
40 REM  *****
50 M2=50
60 M=100
70 DIM N$(M),P$(M),U$(M),C(M),P(M),Q0(M2),N0(M2)
80 INPUT "BID NAME:";B$
90 INPUT "IS THE INPUT FROM A FILE (Y OR N)";A1$
100 IF A1$<>"Y" THEN 140
110 GOSUB 1390 :REM OPEN FILES INPUT
120 GOSUB 730 :REM FILE INPUT
130 GOTO 170
140 GOSUB 320 :REM GET ITEMS
150 GOSUB 480 :REM ACCEPT BID
160 IF C1$="STOP" THEN 230
170 GOSUB 870 :REM PRINT BID
180 PRINT "SHALL I STORE THE BID (Y OR N) "
190 INPUT A2$
200 IF A2$="N" THEN 240
210 GOSUB 1440 :REM OPEN FILES OUTPUT
220 GOSUB 1170 :REM STORE BID
230 REM *****
240 REM  PROGRAM TERMINATION POINT
250 REM *****
260 PRINT
270 PRINT
280 PRINT "PROCESSING COMPLETE"
290 PRINT
300 STOP
310 REM *****
320 REM          GET ITEMS
330 REM *****
340 DATA 1,ITEM 1,EA,1.30,1.70
350 DATA 2,ITEM 2,EA,1.03,1.90
360 DATA 3,ITEM 3,BX,7.49,12.25
370 DATA 4,ITEM 4,EA,.40,.64
380 DATA *
390 I=1
400 READ N$(I)
410 IF N$(I)="*" THEN 450
420 READ P$(I),U$(I),C(I),P(I)
430 I=I+1
440 GOTO 400
450 M=M-1
460 RETURN
470 REM *****
480 REM          ACCEPT BID
490 REM *****
500 PRINT CHR$(147)
```

```

510 J=0
520 M3=0
530 INPUT "CUSTOMER NAME:";C1$
540 IF C1$="STOP" THEN 710
550 INPUT "CUSTOMER ADDRESS:";C2$
560 INPUT "          ";C3$
570 INPUT "TELEPHONE:";T$
580 PRINT
590 REM ***** ADD ENTRIES *****
600 J=J+1
610 N0$=""
620 INPUT "PRODUCT #";N0$
630 IF N0$="" THEN 700
640 GOSUB 1280 :REM FIND ITEM
650 IF P0=0 THEN GOSUB 1590:REM ADD
660 N0(J)=P0
670 PRINT N$(P0);TAB(8);P$(P0);TAB(20);U$(P0);TAB(23);C(P0);TAB(29);P(
P0)
680 INPUT "QUANTITY";Q0(J)
690 GOTO 600
700 M3=J-1
710 RETURN
720 REM *****
730 REM          FILE INPUT
740 REM *****
750 INPUT#1,C1$
760 INPUT#1,C2$,C3$,T$
770 FOR J=1 TO M
780 INPUT#1,N$(J)
790 IF N$(J)="*" THEN 830
800 INPUT#1,P$(J),C(J),P(J),Q0(J)
810 N0(J)=J
820 NEXT J
830 M3=J-1
840 CLOSE 1
850 RETURN
860 REM *****
870 REM          PRINT BID
880 REM *****
890 PRINT CHR$(147)
900 X$="-----"
910 T=0
920 T1=0
930 PRINT C1$
940 PRINT C2$
950 PRINT C3$;TAB(30);T$
960 PRINT
970 PRINT "# ITEM#";TAB(12);"ITEM DESC";TAB(30);"UNIT"
980 PRINT TAB(5);"COST";TAB(15);"PRICE";TAB(25);"QTY";TAB(32);"BID AMT"
990 PRINT X$
1000 FOR I=1 TO M3
1010 IF Q0(I)=0 THEN 1080
1020 PRINT I;N$(N0(I));TAB(10);P$(N0(I));TAB(30);U$(N0(I))
1030 PRINT TAB(5);C(N0(I));TAB(15);P(N0(I));TAB(25);Q0(I);
1040 PRINT TAB(32);P(N0(I))*Q0(I)
1050 PRINT X$
1060 T=T+P(N0(I))*Q0(I)

```

```

1070 T1=T1+C(NO(I))*Q0(I)
1080 NEXT I
1090 PRINT TAB(10);"TOTAL COST";TAB(20);"$";T1
1100 PRINT TAB(10);"TOTAL BID AMT";TAB(28);"$";T
1110 INPUT "CORRECTIONS (Y OR N)";A$
1120 IF A$<>"Y" THEN 1150
1130 GOSUB 1490 :REM CORRECTIONS
1140 GOTO 890
1150 RETURN
1160 REM *****
1170 REM          STORE BID
1180 REM *****
1190 PRINT#2,C1$;R9$;C2$;R9$;C3$;R9$;T$
1200 FOR I=1 TO M3
1210 IF Q0(I)=0 THEN 1230
1220 PRINT#2,N$(NO(I));R9$;P$(NO(I));R9$;C(NO(I));R9$;P(NO(I));R9$;Q0(
I)
1230 NEXT I
1240 PRINT#2,"*"
1250 CLOSE 2
1260 RETURN
1270 REM *****
1280 REM          FIND ITEM
1290 REM *****
1300 P0=0
1310 FOR I=1 TO M
1320 IF N0$=N$(I) THEN 1360
1330 NEXT I
1340 PRINT "ITEM NUMBER NOT FOUND"
1350 GOTO 1370
1360 P0=I
1370 RETURN
1380 REM *****
1390 REM          OPEN FILES INPUT
1400 REM *****
1410 OPEN 1,1,0,B$
1420 RETURN
1430 REM *****
1440 REM          OPEN FILES OUTPUT
1450 R9$=CHR$(13)
1460 OPEN 2,1,1,B$
1470 RETURN
1480 REM *****
1490 REM          CORRECTIONS
1500 REM *****
1510 I=0:INPUT "ENTER # FROM ORDER";I
1520 IF I=0 THEN 1570
1530 IF I<=M3 THEN INPUT "ENTER NEW QTY";Q0(I):GOTO 1510
1540 PRINT "ADD ITEMS"
1550 J=M3
1560 GOSUB 590 :REM ADD ENTRIES
1570 RETURN
1580 REM *****
1590 REM          ADD OTHER ITEMS
1600 REM *****
1610 N$(J)="?????"
1620 INPUT "ENTER ITEM DESC";P$(J)

```


1630 INPUT " UNIT";U\$(J)
 1640 INPUT " COST";C(J)
 1650 INPUT " PRICE";P(J)
 1660 P0=J
 1670 RETURN

RUN
 BID NAME: ? JOHN-DOE
 IS THE INPUT FROM A FILE (Y OR N) ? N

CUSTOMER NAME: ? JOHN F DOE
 CUSTOMER ADDRESS: ? 130 LOCUST WAY
 : ? MIAMI FL 65432
 TELEPHONE: ? 223-4567

PRODUCT # ? 2
 2 ITEM 2 EA 1.03 1.9
 QUANTITY ? 2
 PRODUCT # ? 4
 4 ITEM 4 EA .4 .64
 QUANTITY ? 1
 PRODUCT # ? 2
 2 ITEM 2 EA 1.03 1.9
 QUANTITY ? 2
 PRODUCT # ?

JOHN F DOE
 130 LOCUST WAY
 MIAMI FL 65432 223-4567

#	ITEM#	ITEM DESC	UNIT	COST	PRICE	QTY	BID AMT
1	2	ITEM 2	EA	1.03	1.9	2	3.8
2	4	ITEM 4	EA	.4	.64	1	.64
3	2	ITEM 2	EA	1.03	1.9	2	3.8

TOTAL COST \$ 4.52
 TOTAL BID AMT \$ 8.24
 CORRECTIONS (Y OR N) ? N

SHALL I STORE THE BID (Y OR N)
 ? Y

PRESS RECORD & PLAY ON TAPE
OK

PROCESSING COMPLETE

Break in 300

RUN
BID NAME: ? JOHN-DOE
IS THE INPUT FROM A FILE (Y OR N)? Y
PRESS PLAY ON TAPE
OK

JOHN F DOE
130 LOCUST WAY
MIAMI FL 65432 223-4567

#	ITEM#	ITEM DESC	UNIT	
	COST	PRICE	QTY	BID AMT

1	2	ITEM 2		
	1.03	1.9	2	3.8

2	4	ITEM 4		
	.4	.64	1	.64

3	2	ITEM 2		
	1.03	1.9	2	3.8

TOTAL COST\$ 4.52
TOTAL BID AMT \$ 8.24
CORRECTIONS (Y OR N)? N

SHALL I STORE THE BID (Y OR N)
? N

PROCESSING COMPLETE

Break in 300

MAJOR SYMBOL TABLE — BIDCREAT

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
A1\$	ANSWER VARIABLE
A2\$	ANSWER VARIABLE
B\$	BID NAME
C()	ITEM COST ARRAY
C1\$	CUSTOMER NAME
C2\$	CUSTOMER ADDRESS FIRST LINE
C3\$	CUSTOMER ADDRESS SECOND LINE
I	INDEX AND ARRAY POINTER
J	INDEX AND ARRAY POINTER
M	MAX NUMBER OF BID ITEMS STORED
M2	MAX LINE ITEMS FOR BIDS
M3	NUMBER OF ENTRIES
N\$()	ITEM NUMBER ARRAY
N0\$	INPUT ITEM NUMBER
N0()	POINTER TO BID ITEMS
P\$()	ITEM NAME ARRAY
P()	ITEM PRICE ARRAY
P0	POINTER TO SELECTED ITEM
Q()	QUANTITY ARRAY
R9\$	CARRIAGE RETURN
T	TOTAL BID AMOUNT
T\$	TELEPHONE NUMBER
T1	TOTAL COSTS
U\$()	UNIT FOR BIDDING ARRAY
X\$	LINE OF HYPHENS

FUNCTIONS USED

NAME
CHR\$
CLOSE
DIM
GOSUB
OPEN
RETURN
TAB

6 Children's Work and Play

MPRINT-1

Description

This program produces multiple prints of a single message. It prints any message input to the keyboard as many times as it is requested. It can be used as a learning and familiarization tool for younger children.

Instructions for Use

RUN the program and respond to its requests for information.

Data Entry

All data are entered in response to the program's messages.

Notes and Things to Check

Make sure that no commas are included in the data to be printed and that the number of times to print answer is entered as a number.

Modification Procedures

All program processing is accomplished at lines 130-160 of the program.

```
10 REM          SAVED AT MPRINT-1
20 REM  THIS PROGRAM PROVIDES MULTIPLE
30 REM  LINES OF DATA FROM KEYBOARD
40 REM  INPUT
50 REM  *****
```

```

60 PRINT "ENTER THE DATA TO BE PRINTED"
70 INPUT S$
80 PRINT "HOW MANY TIMES SHALL I PRINT IT";
90 INPUT N
100 PRINT "SHALL I PUT A SPACE BETWEEN EACH";
110 INPUT A$
120 PRINT CHR$(147)
130 FOR I=1 TO N
140 PRINT S$
150 IF A$="Y" THEN PRINT
160 NEXT I
170 REM ** PROGRAM TERMINATION POINT **
180 PRINT
190 PRINT
200 PRINT "PROCESSING COMPLETE"
210 PRINT
220 STOP

```

READY.

```

RUN
ENTER THE DATA TO BE PRINTED
? EXAMPLE DATA TO PRINT 5 TIMES
HOW MANY TIMES SHALL I PRINT IT? 5
SHALL I PUT A SPACE BETWEEN EACH? N

```

```

EXAMPLE DATA TO PRINT 5 TIMES
EXAMPLE DATA TO PRINT 5 TIMES
EXAMPLE DATA TO PRINT 5 TIMES
EXAMPLE DATA TO PRINT 5 TIMES
EXAMPLE DATA TO PRINT 5 TIMES

```

PROCESSING COMPLETE

Break in 220

MAJOR SYMBOL TABLE — MPRINT-1

NAME	DESCRIPTION
A\$	TEMP ANSWER VARIABLE
I	INDEX POINTER
N	NUMBER OF TIMES TO PRINT
S\$	DATA TO PRINT

FUNCTIONS USED

NAME
CHR\$

MPRINT-2

Description

This program produces multiple prints of input supplied as DATA statements. It can be used to interest younger children in the use of the computer and to illustrate a program's operation using DATA statements as input.

Instructions for Use

Replace the data at lines 280-290 with the data to be printed and RUN the program.

Data Entry

All data are entered as DATA statements to the program. These data can take as many lines as you require, but they cannot contain any commas. The first DATA statement tells the computer how many lines of data are to be entered. The sample program outputs contain the following DATA statements:

```
280 DATA 2
290 DATA THIS EXAMPLE DATA (TWO
    LINES OF IT)
295 DATA WILL BE PRINTED 5 TIMES
```

If commas must be included in the data, the entire DATA statement must be enclosed in quotation marks, for example: "A COMMA, IS INCLUDED".

Notes and Things to Check

If the program fails to function correctly, check to make sure that the first DATA statement contains the number of lines of data to be printed and that the remaining data do not contain any commas.

Modification Procedures

The data input and printing functions are performed in the loop at lines 100-180. After each print of the message, the program RESTOREs the data and then prints the message again until the number of prints equals the number requested.

```

10 REM          SAVED AT MPRINT-2
20 REM THIS PROGRAM PROVIDES MULTIPLE
30 REM PRINTS OF DATA FROM DATA STMENTS
40 REM *****
50 PRINT "HOW MANY TIMES SHALL I PRINT IT";
60 INPUT N
70 PRINT "SHALL I PUT A SPACE BETWEEN EACH";
80 INPUT A$
90 PRINT CHR$(147)
100 FOR I=1 TO N
110 READ N1
120 FOR J=1 TO N1
130 READ S$
140 PRINT S$
150 NEXT J
160 IF A$="Y" THEN PRINT
170 RESTORE
180 NEXT I
190 REM ** PROGRAM TERMINATION POINT **
200 PRINT
210 PRINT
220 PRINT "PROCESSING COMPLETE"
230 PRINT
240 STOP
250 REM *****
260 REM   DATA STATEMENTS FOR PRINT
270 REM *****
280 DATA      :REM  REPLACE WITH # OF LINES OF DATA
290 DATA      :REM  REPLACE WITH DATA TO BE PRINTED
300 END

```

```

RUN
HOW MANY TIMES SHALL I PRINT IT? 3
SHALL I PUT A SPACE BETWEEN EACH? Y

THIS EXAMPLE DATA ( TWO LINES OF IT)
WILL BE PRINTED 5 TIMES.

THIS EXAMPLE DATA ( TWO LINES OF IT)
WILL BE PRINTED 5 TIMES.

THIS EXAMPLE DATA ( TWO LINES OF IT)
WILL BE PRINTED 5 TIMES.

```

PROCESSING COMPLETE

Break in 240

MAJOR SYMBOL TABLE — MPRINT-2

NAME	DESCRIPTION
A\$	TEMP ANSWER VARIABLE
I	INDEX POINTER
J	INDEX
N	NUMBER OF TIMES TO PRINT
N1	NUMBER OF DATA STATEMENTS
S\$	DATA TO PRINT

FUNCTIONS USED

NAME
CHR\$

EXPENSE

Description

This program provides an ideal method for a younger child to become familiar both with handling finances and with using a computer. It provides a child's version of an income statement, which can be used to stimulate financial responsibility and encourage computer awareness.

Instructions for Use

The data are entered as DATA statements before the program is run. The resulting data should be resaved to tape or disk for use during the next week's processing. The program can then be RUN and its results printed. The program also reminds the child to enter current income and expenses before running it.

Data Entry

Data are entered as DATA statements in the following form:

1. The first entry must contain the number of expense categories that will be entered.
2. The second entry must contain a name for each of these expense categories.
3. The remaining entries contain date, income, and then one amount for each of the expense categories that are specified. One record is entered for each week and each new record should be added at the end of all preceding records.
4. The last DATA statement must contain STOP to signify the end of data to the program.

Notes and Things to Check

You may wish to assist the child in setting up the initial DATA statements, and in determining the expense categories to be included. If program errors occur they will probably be as a result of a missing data item or the presence (or lack of), a comma in the data.

Modification Procedures

This program can easily be modified to provide additional information or different printing formats. All data entries are made to arrays (the "DATA ENTRY" subroutine), which can then be manipulated in any desired way. The actual computation and printing of the data does not occur until the "COMPUTE DISPLAY" subroutine is executed at lines

370-610. The computation of savings is also accomplished in a separate subroutine. This routine uses the INT function to round the data to dollars and cents. This is required to insure correct formatting.

```

10 REM          SAVED AT EXPENSE
20 REM THIS PROGRAM IS FOR CALCULATING
30 REM A CHILD'S BUDGET/EXPENSES
40 REM *****
50 X9$="*****"
60 PRINT "DID YOU REMEMBER TO PUT YOUR EXPENSES IN THE DATA"
70 INPUT A$
80 IF A$="Y" THEN 110
90 PRINT "ENTER YOUR EXPENSES IN A DATA STATEMENT AND THEN RUN AGAIN"
100 GOTO 140
110 GOSUB 200      :REM DATA ENTRY
120 GOSUB 370     :REM COMPUTE DISPLAY
130 REM *****
140 REM  PROGRAM TERMINATION POINT
150 REM *****
160 PRINT
170 PRINT "PROCESSING COMPLETE"
180 STOP
190 REM *****
200 REM          DATA ENTRY
210 REM *****
220 READ N      :REM # OF EXP CATEGORIES
230 FOR I=1 TO N
240 READ E$(I)
250 NEXT I
260 READ D$     :REM READ DATE (OR STOP)
270 IF D$="STOP" THEN 350
280 READ A      :REM INCOME AMOUNT
290 A0=A0+A
300 FOR I=1 TO N
310 READ E(I) :REM EXPENSE AMOUNTS
320 T(I)=T(I)+E(I)
330 NEXT I
340 GOTO 260
350 RETURN
360 REM *****
370 REM          COMPUTE DISPLAY
380 REM *****
390 PRINT CHR$(147)
400 PRINT X9$
410 PRINT "  CURRENT WEEK";TAB(20);"PAST"
420 PRINT
430 PRINT "INCOME";TAB(10);A;TAB(20);A0
440 PRINT
450 PRINT "EXPENSES"
460 FOR I=1 TO N
470 PRINT E$(I);TAB(10);E(I);TAB(20);T(I)
480 E0=E0+E(I)
490 T0=T0+T(I)
500 NEXT I

```

```

510 PRINT TAB(10);"-----";TAB(20);"-----"
520 PRINT "TOT EXP";TAB(10);"$";E0;TAB(20);"$";T0
530 PRINT X9$
540 PRINT "    THIS WEEK";TAB(20)"PAST"
550 PRINT
560 GOSUB 730 :REM COMPUTE SAVINGS
570 PRINT "INCOME";TAB(10);"$";A;TAB(20);"$";A0
580 PRINT "EXPENSES";TAB(10);"$";E0;TAB(20);"$";T0
590 PRINT "-----"
600 PRINT "SAVINGS";TAB(10);"$";S;TAB(20);"$";S0
610 PRINT X9$
620 REM *****
630 REM          DATA STATEMENTS
640 REM *****
650 DATA 5
660 DATA MISC,PENCILS,GUM,BOOKS,LUNCH
670 DATA WEEK 1
680 DATA 1,.1,.2,.3,.4,.5
690 DATA WEEK 2
700 DATA 2,.1,.3,.4,.5,.6
710 DATA STOP
720 REM *****
730 REM          COMPUTE SAVINGS
740 REM *****
750 S=A-E0
760 S0=A0-T0
770 S=INT(S*100+5/1000)/100
780 S0=INT(S0*100+5/1000)/100
790 RETURN

```

```

RUN
DID YOU REMEMBER TO PUT YOUR EXPENSES IN
THE DATA
? Y

```

```

*****

```

	CURRENT WEEK	PAST
INCOME	2	3
EXPENSES		
MISC	.1	.2
PENCILS	.3	.5
GUM	.4	.7
BOOKS	.5	.9
LUNCH	.6	1.1
TOT EXP	\$ 1.9	\$ 3.4

```

*****

```

	THIS WEEK	PAST
INCOME	\$ 2	\$ 3
EXPENSES	\$ 1.9	\$ 3.4

SAVINGS	\$.1	\$-.4

PROCESSING COMPLETE
 Break in 180

MAJOR SYMBOL TABLE — EXPENSE

NAME	DESCRIPTION
A0	TOTAL PAST INCOME
D\$	DATE
E()	ARRAY OF EXPENSE AMOUNTS
E0	TOTAL EXPENSES
I	INDEX POINTER
N	NUMBER OF EXPENSES
N\$()	ARRAY OF EXPENSE NAMES
S	SAVINGS THIS WEEK
S0	TOTAL SAVINGS
T()	TOTAL ARRAY
T0	TOTAL PAST EXPENSES
X9\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
CHR\$
GOSUB
INT
RETURN
TAB

CODER

Description

This program codes and decodes messages. It allows friends to send one another messages that can be decoded only through the use of this program by someone who knows what code number was used.

Instructions for Use

RUN the program and respond to its requests. Do not overlook the requirement that you enter a single asterisk as the last line of a message (see example output).

Data Entry

All data are entered in response to the program's requests.

Notes and Things to Check

Positioning of the characters in all messages is critical to correct decoding. It is therefore important to enter all messages (including all spaces and punctuation) very carefully. For a message to be decoded correctly, the same code formula must be used in coding and decoding.

Modification Procedures

The nature of this program's function makes an explanation of its processing procedures inappropriate.

```
10 REM      SAVED AT CODER
20 REM      THIS PROGRAM CODES AND DECODES
30 REM      MESSAGES
40 REM      *****
50 M0=10
60 A0=38
70 DIM A$(A0),M$(M0),M1$(M0)
80 DATA A,B,C,D,E,F,G,H,I,J,K,L,M,N,O,P,Q,R,S,T,U,V,W,X,Y,Z,.
90 DATA 1,2,3,4,5,6,7,8,9,0,*
100 FOR I=1 TO A0
110 READ A$(I)
120 NEXT I
130 PRINT "ENTER CODE FORMULA (1-999)";
140 INPUT F$
150 L=LEN(F$)
160 IF L>3 THEN L=3
170 FOR I=1 TO L
180 F(I)=VAL(MID$(F$,I,1))
```

```

190 NEXT I
200 PRINT "SHALL I CODE A MESSAGE (Y OR N)"
210 INPUT A0$
220 IF A0$="Y" THEN GOSUB 390 :REM MESSAGE ENTRY
230 F2=0
240 F1=0
250 IF A0$="Y" THEN GOSUB 1110 :REM PRINT MESSAGE
260 PRINT "SHALL I DECODE A MESSAGE (Y OR N)"
270 INPUT A0$
280 IF A0$="Y" THEN GOSUB 750 :REM CODED MESSAGE ENTRY

290 IF A0$="Y" THEN GOSUB 1110 :REM PRINT MESSAGE
300 REM *****
310 REM PROGRAM TERMINATION POINT
320 REM *****
330 PRINT
340 PRINT
350 PRINT "PROCESSING COMPLETE"
360 PRINT
370 STOP
380 REM *****
390 REM MESSAGE ENTRY
400 REM *****
410 PRINT "ENTER YOUR MESSAGE NOW"
420 PRINT "YOUR LAST ENTRY MUST BE *"
430 FOR I=1 TO M0
440 INPUT M$(I)
450 IF M$(I)="*" THEN 490
460 L=LEN(M$(I))
470 GOSUB 520 :REM CODE MESSAGE
480 NEXT I
490 I0=I-1
500 RETURN
510 REM *****
520 REM CODE MESSAGE
530 REM *****
540 M1$(I)=M$(I)
550 FOR J=1 TO L
560 F1=F1+F(2)
570 IF F1>A0 THEN F1=(F1-A0)+F(3)
580 IF F1>A0 THEN 570
590 FOR K=1 TO A0
600 IF MID$(M$(I),J,1)=A$(K) THEN 630
610 NEXT K
620 GOTO 690
630 F2=K+F(1)+F1
640 IF F2<=A0 THEN 670
650 F2=F2-A0
660 GOTO 640
670 M1$(I)=LEFT$(M1$(I),J-1)+A$(F2)
680 GOTO 700
690 M1$(I)=LEFT$(M1$(I),J-1)+MID$(M$(I),J,1)
700 NEXT J
710 REM PRINT M$(I) WILL PRINT THE INPUT HERE
720 REM PRINT M1$(I) WILL PRINT THE OUTPUT HERE
730 RETURN
740 REM *****

```

```

750 REM          CODED MESSAGE ENTRY
760 REM *****
770 PRINT "ENTER CODED MESSAGE NOW"
780 PRINT "YOUR LAST ENTRY MUST BE  * "
790 FOR I=1 TO M0
800 INPUT M$(I)
810 IF M$(I)="*" THEN 850
820 L=LEN(M$(I))
830 GOSUB 870 :REM DECODE MESSAGE
840 NEXT I
850 I0=I-1
860 RETURN
870 REM *****
880 REM          DECODE MESSAGE
890 REM *****
900 M1$(I)=M$(I)
910 FOR J=1 TO L
920 F1=F1+F(2)
930 IF F1>A0 THEN F1=(F1-A0)+F(3)
940 IF F1>A0 THEN 930
950 FOR K=1 TO A0
960 IF MID$(M$(I),J,1)=A$(K) THEN 990
970 NEXT K
980 GOTO 1050
990 F2=K-F(1)-F1 :REM *
1000 IF F2>=0 THEN 1030 :REM 0
1010 F2=F2+A0 :REM *
1020 GOTO 1000
1030 M1$(I)=LEFT$(M1$(I),J-1)+A$(F2)
1040 GOTO 1060
1050 M1$(I)=LEFT$(M1$(I),J-1)+MID$(M$(I),J,1)
1060 NEXT J
1070 REM PRINT M$(I) WILL PRINT INPUT MESSAGE HERE
1080 REM PRINT M1$(I) WILL PRINT OUTPUT MESSAGE HERE
1090 RETURN
1100 REM *****
1110 REM          PRINT MESSAGES
1120 REM *****
1130 PRINT "SHALL I PRINT THE OUTPUT MESSAGE(Y OR N) "
1140 INPUT A$
1150 IF A$="N" THEN 1250
1160 PRINT
1170 FOR I=1 TO I0
1180 PRINT M1$(I)
1190 NEXT I
1200 PRINT
1210 PRINT "SHALL I PRINT THE CODE FORMULA-#(Y OR N) "
1220 INPUT A$
1230 IF A$="Y" THEN PRINT F$
1240 PRINT
1250 PRINT "SHALL I PRINT THE INPUT MESSAGE(Y OR N) "
1260 INPUT A$
1270 IF A$="N" THEN 1360
1280 PRINT
1290 FOR I=1 TO I0
1300 PRINT M$(I)
1310 NEXT I

```

```
1320 PRINT
1330 PRINT "SHALL I PRINT THE CODE FORMULA-NUMBER (Y OR N)"
1340 INPUT A$
1350 IF A$="Y" THEN PRINT F$
1360 RETURN
```

```
RUN
ENTER CODE FORMULA (1-999)? 235
SHALL I CODE A MESSAGE (Y OR N)
? Y
ENTER YOUR MESSAGE NOW
YOUR LAST ENTRY MUST BE *
? HI READER.
? THIS MESSAGE WILL BE ENCODED.
? *
SHALL I PRINT THE OUTPUT MESSAGE(Y OR N)
? Y
```

```
MQ 5VU.4IU
QH03 3YDG3AB 4TZ2 Y4 OKCWOSUI
```

```
SHALL I PRINT THE CODE FORMULA #(Y OR N)
? N
```

```
SHALL I PRINT THE INPUT MESSAGE(Y OR N)
? N
SHALL I DECODE A MESSAGE (Y OR N)
? N
```

PROCESSING COMPLETE

Break in 370

```
RUN
ENTER CODE FORMULA (1-999)? 235
SHALL I CODE A MESSAGE (Y OR N)
? N
SHALL I DECODE A MESSAGE (Y OR N)
? Y
ENTER CODED MESSAGE NOW
YOUR LAST ENTRY MUST BE *
? MQ 5VU.4IU
? QH03 3YDG3AB 4TZ2 Y4 OKCWOSUI
? *
SHALL I PRINT THE OUTPUT MESSAGE(Y OR N)
? Y
```

```
HI READER.
THIS MESSAGE WILL BE ENCODED.
```


SHALL I PRINT THE CODE FORMULA #(Y OR N)
 ? Y
 235

SHALL I PRINT THE INPUT MESSAGE(Y OR N)
 ? Y

MQ 5VU.4IU
 QHQ3 3YDG3AB 4TZ2 Y4 OKCWSUI

SHALL I PRINT THE CODE FORMULA-NUMBER (Y OR N)
 ? N

PROCESSING COMPLETE

Break in 370

MAJOR SYMBOL TABLE — CODER

NAME	DESCRIPTION
A\$()	ALPHABET OF CHARS TO CODE
AO	NUMBER OF CHARS TO CODE
AO\$	ANSWER VARIABLE
F\$	STRING OF FORMULA
F (1)	START FACTOR FOR CODE
F (2)	INCREMENT FACTOR FOR CODE
F(3)	RESET INCREMENT FOR CODE
F1	CALCULATED NUMBER
F2	LETTER TO GET
10	NUMBER OF MESSAGES INPUT
L	LENGTH OF DATA
M\$()	INPUT MESSAGE
MO	MAX NUMBER OF MESSAGES ALLOWED
M1\$()	OUTPUT MESSAGE
N	NUMBER OF CHARS IN MESSAGE

FUNCTIONS USED

NAME
DIM
GOSUB
LEFT\$
LEN
MID\$
RETURN
VAL

N-PAPER

Description

This program computes a newspaper delivery person's weekly bills and earnings. It can be used to help control the financial aspects of newspaper delivery and to familiarize the user with a computer.

Instructions for Use

Before you use the program you should change the cost and price factors at lines 70-100 to the correct values for your own newspaper. The following changes should be made:

Line 70 Set C0 = to your cost for each daily paper
Line 80 Set C7 = to your cost for each Sunday paper
Line 90 Set P0 = to your customer's price for each daily delivery
Line 100 Set P7 = to your customer's price for each Sunday delivery.

Each week thereafter, you can RUN the program and compute your bills and earnings. Enter the number of deliveries in response to the program's requests.

Data Entry

Cost and price factors are entered directly within the program. See the instructions for lines 70-100. All other data, such as the number of deliveries, are entered in response to program requests.

Notes and Things to Check

Internal storage formats of numbers within the Commodore 64 (and other computers) may cause numbers to be displayed with many decimal positions. Your projected bills and earnings may be distorted slightly by this. If you receive additional income for such jobs as "stuffing" advertisements, it is not accounted for in this program.

Modification Procedures

This program contains comments on all aspects of its processing. Before modifying it you should review the listing to determine the location of the individual computations. If your newspaper includes income for items such as the advertising "stuffing" mentioned above, you may want to include these in the keyboard input and add this income to your totals.

```

10 REM      SAVED AT N-PAPER
20 REM      PROGRAM TO COMPUTE BILL AND
30 REM      EARNINGS FOR NEWSPAPER DELIVERY
40 REM      *****
50 REM      THE FOLLOWING FACTORS MUST BE
60 REM      CHANGED TO YOU PAPER'S NUMBERS
70 C0=.1
80 C7=.55 :REM COST FOR DELIVERY SUNDAY
90 P0=.15 :REM PRICE FOR DAILY DELIVERY
100 P7=.7 :REM PRICE FOR SUNDAY DELIVERY
110 REM      *****
120 PRINT "ENTER THE NUMBER OF DAILY DELIVERIES"
130 INPUT "      MONDAY";D1
140 INPUT "      TUESDAY";D2
150 INPUT "      WEDNESDAY";D3
160 INPUT "      THURSDAY";D4
170 INPUT "      FRIDAY";D5
180 INPUT "      SATURDAY";D6
190 INPUT "NUMBER OF SUNDAY DELIVERIES";D7
200 REM *CALCULATE PRICE AND EARNINGS*
210 D0=D1+D2+D3+D4+D5+D6 :REM # DAILIES
220 T0=D0*C0 :REM TOTAL DAILY COSTS
230 T7=D7*C7 :REM TOTAL SUNDAY COSTS
240 B0=D0*P0 :REM TOT PRICE DAILIES
250 B7=D7*P7 :REM TOT PRICE SUNDAYS
260 PRINT "*****"
270 PRINT "TOTAL PRICE (COLLECTIONS) "
280 PRINT D0;TAB(6);"DAILIES";TAB(15);B0
290 PRINT D7;TAB(6);"SUNDAYS";TAB(15);B7;TAB(22);"TOTAL";TAB(30);B0+B7
300 PRINT "TOTAL COSTS (BILL) "
310 PRINT TAB(6);"DAILIES";TAB(15);T0
320 PRINT TAB(6);"SUNDAYS";TAB(15);T7;TAB(22);"TOTAL";TAB(30);T0+T7
330 PRINT "TOTAL EARNINGS"
340 PRINT TAB(6);"DAILIES";TAB(15);B0-T0
350 PRINT TAB(6);"SUNDAYS";TAB(15);B7-T7;TAB(22);"TOTAL";TAB(30);B0+B7-T0-T7
360 PRINT "*****"
370 STOP

```

RUN

ENTER THE NUMBER OF DAILY DELIVERIES

MONDAY? 10

TUESDAY? 10

WEDNESDAY? 10

THURSDAY? 10

FRIDAY? 10

SATURDAY? 10

NUMBER OF SUNDAY DELIVERIES? 20

TOTAL PRICE (COLLECTIONS)

60 DAILIES 9

20 SUNDAYS 14 TOTAL 23

TOTAL COSTS (BILL)

DAILIES 6

SUNDAYS 11 TOTAL 17

TOTAL EARNINGS
 DAILIES 3
 SUNDAYS 3
 TOTAL 6

 Break in 370

MAJOR SYMBOL TABLE — N-PAPER

NAME	DESCRIPTION
B0	TOTAL PRICE OF DAILIES
B7	TOTAL PRICE OF SUNDAYS
C0	COST OF DAILY PAPER
C7	COST OF SUNDAY PAPER
D0	TOTAL DAILY DELIVERIES
D1	MONDAY DELIVERIES
D2	TUESDAY DELIVERIES
D3	WEDNESDAY DELIVERIES
D4	THURSDAY DELIVERIES
D5	FRIDAY DELIVERIES
D6	SATURDAY DELIVERIES
D7	SUNDAY DELIVERIES
P0	PRICE OF DAILY
P7	PRICE OF SUNDAY
T0	TOTAL DAILY COSTS
T7	TOTAL SUNDAY COSTS

FUNCTIONS USED

NAME
TAB

7 Schoolwork

FLASH-CD

Description

This program is designed to help grade school children (and others) to perform addition, subtraction, multiplication, and division. It provides computer-assisted flash cards that check their answers and summarize their accuracy. It can be used to help children overcome difficulties or increase their ability to work with numbers and solve arithmetic problems.

Instructions for Use

You probably will want to RUN the program a few times to familiarize yourself with its operation before you have your children use it, and you will probably want to help them in RUNning it the first few times they use it. During the actual RUN of the program, you are asked to choose the type of arithmetic you want to practice and the size of the data to be presented. Each of these items is entered in response to a question asked by the computer. The user should then enter the answer and press the RETURN key, whenever the "ANSWER?" message appears on the screen. All answers are shown as whole numbers, except those for the division problems which call for answers correct to two decimal positions.

Data Entry

Not applicable to this program.

Notes and Things to Check

This program operates correctly when proper answers are given to its questions. If the program fails to produce expected results, it is possible that one of the answers to the initial questions was given incorrectly. In addition, it is important to emphasize to a child using the program that the RETURN key must be pressed after each entry. Always remember that the division problems require an answer correct to two decimal positions.

Modification Procedures

Each of the arithmetic processes is separated into its own subroutine to facilitate changes for that process only. They all share the common functions of setting the number size, checking the answer, and displaying the results. If you want to change the number of decimal points expected in the division problems, you can do so by modifying the factors included in the statement at line #2170. If, for example, you would like division answers to be correct to one decimal position, change the statement as follows:

```
2170 IF INT (A0*10)/10 < > A THEN 2210
```

```
10 REM          SAVED AT FLASH-CD
20 REM  PROGRAM TO CREATE FLASH CARDS
30 REM  FOR EDUCATIONAL USE
40 REM  *****
50 PRINT
60 S=0
70 PRINT CHR$(147)
80 PRINT
90 PRINT "THIS PROGRAM CREATES FLASH CARDS"
100 PRINT "HOW MANY FLASH CARDS WOULD YOU LIKE"
110 INPUT T
120 PRINT "DO YOU WISH TO CREATE FLASH CARDS FOR:"
130 PRINT "  ADDITION          - ENTER A"
140 PRINT "  SUBTRACTION         - ENTER S"
150 PRINT "  MULTIPLICATION       - ENTER M"
160 PRINT "  DIVISION             - ENTER D"
170 PRINT
180 PRINT "  ENTER YOUR CHOICE";
190 INPUT A$
200 PRINT
210 IF A$="A" THEN GOSUB 350 :REM ADD
220 IF A$="S" THEN GOSUB 650 :REM SUB
230 IF A$="M" THEN GOSUB 1020:REM MULT
240 IF A$="D" THEN GOSUB 1370:REM DIV
250 GOSUB 1930 :REM DISPLAY RESULTS
260 REM *****
270 REM  PROCESSING COMPLETE
280 REM *****
290 PRINT
300 PRINT
```

```

310 PRINT "PROCESSING COMPLETE"
320 PRINT
330 STOP
340 REM *****
350 REM          ADDITION
360 REM *****
370 PRINT "HOW LARGE SHOULD THE NUMBERS BE:"
380 PRINT TAB(5);"OPTION 1.....LESS THAN 10"
390 PRINT TAB(12);"2.....BETWEEN 10 AND 100"
400 PRINT TAB(12);"3.....BETWEEN 100 AND 1000""
410 PRINT TAB(12);"4.....GREATER THAN 1000"
420 PRINT
430 PRINT "ENTER YOUR CHOICE";
440 INPUT O
450 IF O > 0 AND O < 5 THEN 480
460 PRINT "YOU MUST ENTER A NUMBER 1, 2, 3, OR 4 TO CHOOSE THE CORRECT
OPTION"
470 GOTO 370
480 PRINT "PRESS THE RETURN KEY TO START";
490 GOSUB 2270 :REM SET NUMBER SIZE
500 PRINT CHR$(147)
510 R1=INT(RND(S)*M0)+M1
520 R2=INT(RND(S)*M0)+M1
530 N=N+1
540 PRINT "QUESTION # ";N
550 PRINT TAB(20);R1
560 PRINT TAB(19);"+";R2
570 PRINT TAB(18);"-----"
580 PRINT
590 INPUT "ANSWER ";A
600 A0=R1+R2
610 GOSUB 2160 :REM CHECK ANSWER
620 IF N < T THEN 510
630 RETURN
640 REM *****
650 REM          SUBTRACTION
660 REM *****
670 PRINT "HOW LARGE SHOULD THE NUMBERS BE"
680 PRINT TAB(5);"OPTION 1.....LESS THAN 10"
690 PRINT TAB(12);"2.....BETWEEN 10 AND 100"
700 PRINT TAB(12);"3.....BETWEEN 100 AND 1000"
710 PRINT TAB(12);"4.....GREATER THAN 1000"
720 PRINT
730 PRINT "ENTER YOUR CHOICE";
740 INPUT O
750 IF O > 0 AND O < 5 THEN 780
760 PRINT "YOU MUST ENTER A NUMBER 1, 2, 3, OR 4 TO CHOOSE THE CORRECT
OPTION"
770 GOTO 370
780 PRINT "WILL YOU ALLOW NEGATIVE ANSWERS (Y OR N)":INPUT A$
790 PRINT "PRESS THE RETURN KEY TO START";
800 GOSUB 2270 :REM SET NUMBER SIZE
810 PRINT CHR$(147)
820 R1=INT(RND(S)*M0)+M1
830 R2=INT(RND(S)*M0)+M1
840 R3=R1
850 R4=R2

```

```

860 IF A$="Y" THEN 900
870 IF R2 <= R1 THEN 900
880 R3=R2
890 R4=R1
900 N=N+1
910 PRINT "QUESTION # ";N
920 PRINT TAB(20);R3
930 PRINT TAB(19)"-";R4
940 PRINT TAB(18);"-----"
950 PRINT
960 INPUT "ANSWER ";A
970 A0=R3-R4
980 GOSUB 2160 :REM CHECK ANSWER
990 IF N < T THEN 820
1000 RETURN
1010 REM *****
1020 REM          MULTIPLICATION
1030 REM *****
1040 PRINT "HOW LARGE SHOULD THE NUMBERS BE"
1050 PRINT TAB(5);"OPTION 1.....LESS THAN 10"
1060 PRINT TAB(12);"2.....BETWEEN 10 AND 100"
1070 PRINT TAB(12);"3.....BETWEEN 100 AND 1000"
1080 PRINT TAB(12);"4.....GREATER THAN 1000"
1090 PRINT
1100 PRINT "ENTER YOUR CHOICE";
1110 INPUT O
1120 IF O > 0 AND O < 5 THEN 1150
1130 PRINT "YOU MUST ENTER A NUMBER 1,2,3, OR 4 TO CHOOSE THE CORRECT
OPTION"
1140 GOTO 1250
1150 PRINT "PRESS THE RETURN KEY TO START";
1160 GOSUB 2270 :REM SET NUMBER SIZE
1170 PRINT CHR$(147)
1180 R1=INT(RND(S)*M0)+M1
1190 R2=INT(RND(S)*M0)+M1
1200 R3=R1
1210 R4=R2
1220 IF R1>R2 THEN 1250
1230 R3=R2
1240 R4=R1
1250 N=N+1
1260 PRINT "QUESTION # ";N
1270 PRINT TAB(20);R1
1280 PRINT TAB(19)"X";R2
1290 PRINT TAB(18);"-----"
1300 PRINT
1310 INPUT "ANSWER ";A
1320 A0=R1*R2
1330 GOSUB 2160 :REM CHECK ANSWER
1340 IF N < T THEN 1180
1350 RETURN
1360 REM *****
1370 REM          DIVISION
1380 REM *****
1390 PRINT "HOW LARGE SHOULD THE DIVISOR BE:"
1400 PRINT TAB(5);"OPTION 1.....LESS THAN 10"
1410 PRINT TAB(12);"2.....BETWEEN 10 AND 100"

```



```

1420 PRINT TAB(12);"3.....BETWEEN 100 AND 1000"
1430 PRINT TAB(12);"4.....GREATER THAN 1000"
1440 PRINT
1450 PRINT "ENTER YOUR CHOICE";
1460 INPUT O1
1470 IF O1 > 0 AND O1 < 5 THEN 1500
1480 PRINT "YOU MUST ENTER A NUMBER 1,2,3, OR 4 TO CHOOSE THE CORRECT
OPTION"
1490 GOTO 1390
1500 IF O1=1 THEN 1550
1510 IF O1=2 THEN 1580
1520 IF O1=3 THEN 1610
1530 IF O1=4 THEN 1640
1540 GOTO 1660
1550 M2=10
1560 M3=0
1570 GOTO 1660
1580 M2=89
1590 M3=10
1600 GOTO 1660
1610 M2=899
1620 M3=100
1630 GOTO 1660
1640 M2=8999
1650 M3=1000
1660 PRINT "HOW LARGE SHOULD THE DIVIDEND BE:"
1670 PRINT TAB(5);"OPTION 1.....LESS THAN 10"
1680 PRINT TAB(12);"2.....BETWEEN 10 AND 100"
1690 PRINT TAB(12);"3.....BETWEEN 100 AND 1000
1700 PRINT TAB(12);"4.....GREATER THAN 1000"
1710 PRINT
1720 PRINT "ENTER YOUR CHOICE";
1730 INPUT O
1740 IF O > 0 AND O < 5 THEN 1770
1750 PRINT "YOU MUST ENTER A NUMBER 1,2,3, OR 4 TO CHOOSE THE CORRECT
OPTION"
1760 GOTO 1650
1770 PRINT "PRESS THE RETURN KEY TO START";
1780 GOSUB 2270 :REM SET NUMBER SIZE
1790 PRINT CHR$(147)
1800 R1=INT(RND(S)*M0)+M1
1810 R2=INT(RND(S)*M2)+M3
1820 IF R2=0 THEN 1800
1830 N=N+1
1840 PRINT "QUESTION # ";N
1850 PRINT TAB(23+O1);"_____"
1860 PRINT TAB(20);R2;"/";R1
1870 PRINT
1880 INPUT "ANSWER ";A
1890 A0=R1/R2
1900 GOSUB 2160 :REM CHECK ANSWER
1910 IF N < T THEN 1800
1920 RETURN
1930 REM *****
1940 REM DISPLAY RESULTS
1950 REM *****
1960 P1=INT((C/T+5/1000)*100)

```

```

1970 P2=INT((W/T+5/1000)*100)
1980 PRINT "*****"
1990 PRINT TAB(5);"THAT'S THE LAST ONE"
2000 PRINT
2010 PRINT "PRESS RETURN TO"
2020 PRINT "SEE HOW YOU DID";
2030 INPUT A$
2040 PRINT CHR$(147)
2050 PRINT "*****"
2060 PRINT TAB(5);"SCORE BOARD"
2070 PRINT
2080 PRINT "NUMBER CORRECT...";C;TAB(25);"PERCENT";P1
2090 PRINT "NUMBER INCORRECT..";W;TAB(25);"PERCENT";P2
2100 PRINT "TOTAL NUMBER.....";T
2110 PRINT
2120 PRINT "*****"
2130 RETURN
2140 REM *****
2150 REM          CHECK ANSWER
2160 REM *****
2170 IF INT(A0*100)/100<>A THEN 2210
2180 PRINT "CORRECT"
2190 C=C+1
2200 GOTO 2240
2210 PRINT "INCORRECT"
2220 PRINT "THE CORRECT ANSWER IS ";A0
2230 W=W+1
2240 PRINT
2250 RETURN
2260 REM *****
2270 REM          SET NUMBER SIZE
2280 INPUT A9$:REM WAIT FOR GOAHEAD
2290 IF O=1 THEN 2340
2300 IF O=2 THEN 2370
2310 IF O=3 THEN 2400
2320 IF O=4 THEN 2430
2330 GOTO 2450
2340 M0=10
2350 M1=0
2360 GOTO 2450
2370 M0=89
2380 M1=10
2390 GOTO 2450
2400 M0=899
2410 M1=100
2420 GOTO 2450
2430 M0=8999
2440 M1=1000
2450 RETURN

```

RUN

THIS PROGRAM CREATES FLASH CARDS
HOW MANY FLASH CARDS WOULD YOU LIKE
? 2

DO YOU WISH TO CREATE FLASH CARDS FOR:

ADDITION - ENTER A
SUBTRACTION - ENTER S
MULTIPLICATION - ENTER M
DIVISION - ENTER D

ENTER YOUR CHOICE? A

HOW LARGE SHOULD THE NUMBERS BE:

OPTION 1.....LESS THAN 10
2.....BETWEEN 10 AND 100
3.....BETWEEN 100 AND 1000
4.....GREATER THAN 1000

ENTER YOUR CHOICE? 1

PRESS THE RETURN KEY TO START?

QUESTION # 1

$$\begin{array}{r} 4 \\ + 3 \\ \hline \end{array}$$

ANSWER ? 6

INCORRECT

THE CORRECT ANSWER IS 7

QUESTION # 2

$$\begin{array}{r} 4 \\ + 1 \\ \hline \end{array}$$

ANSWER ? 5

CORRECT

THAT'S THE LAST ONE

PRESS RETURN TO
SEE HOW YOU DID?

SCORE BOARD

NUMBER CORRECT.... 1 PERCENT 50
NUMBER INCORRECT.. 1 PERCENT 50
TOTAL NUMBER..... 2

PROCESSING COMPLETE

Break in 330

RUN

THIS PROGRAM CREATES FLASH CARDS
HOW MANY FLASH CARDS WOULD YOU LIKE
? 2

DO YOU WISH TO CREATE FLASH CARDS FOR:
ADDITION - ENTER A
SUBTRACTION - ENTER S
MULTIPLICATION - ENTER M
DIVISION - ENTER D

ENTER YOUR CHOICE? D

HOW LARGE SHOULD THE DIVISOR BE:
OPTION 1.....LESS THAN 10
2.....BETWEEN 10 AND 100
3.....BETWEEN 100 AND 1000
4.....GREATER THAN 1000

ENTER YOUR CHOICE? 1

HOW LARGE SHOULD THE DIVIDEND BE:
OPTION 1.....LESS THAN 10
2.....BETWEEN 10 AND 100
3.....BETWEEN 100 AND 1000
4.....GREATER THAN 1000

ENTER YOUR CHOICE? 2
PRESS THE RETURN KEY TO START?

QUESTION # 1

3 / -----
33

ANSWER ? 11
CORRECT

QUESTION # 2

4 / -----
94

ANSWER ? 29

INCORRECT
THE CORRECT ANSWER IS 23.5

THAT'S THE LAST ONE

PRESS RETURN TO
SEE HOW YOU DID?

SCORE BOARD

NUMBER CORRECT.... 1 PERCENT 50
NUMBER INCORRECT.. 1 PERCENT 50
TOTAL NUMBER..... 2

PROCESSING COMPLETE

Break in 330

MAJOR SYMBOL TABLE — FLASH-CD

NAME	DESCRIPTION
A	ANSWER INPUT
A\$	ANSWER VARIABLE
AO	CORRECT ANSWER
C	NUMBER CORRECT
MO	SCALING FACTOR
M1	SCALING FACTOR
M2	SCALING FACTOR
M3	SCALING FACTOR
N	NUMBER OF PROBLEMS PRESENTED
O	OPTION NUMBER
O1	OPTION NUMBER
P1	PERCENT CORRECT
P2	PERCENT WRONG
R1	RANDOM NUMBER
R2	RANDOM NUMBER
S	RANDOM SEED
T	TOTAL QUESTIONS
W	NUMBER WRONG

FUNCTIONS USED

NAME
CHR\$
GOSUB
INT
RETURN
RND
TAB

W-SOUNDS

Description

This program enables you to use the computer in teaching reading skills to your younger children. It allows you to enter the word sounds you are teaching your child and then displays the words in a dynamic format as you are sounding them out. It can assist in stimulating children to develop early reading and word-sound recognition skills.

Instructions for Use

RUN the program and then enter the sounds in response to the program's messages. To teach the sound and recognition of the syllable *ig*, you can use the sample run. At your request, the program will display parts of words on the screen separated by several spaces. For example, the P will appear on the left side of the screen and the IG will appear on the right. The IG will then move gradually across the screen until it forms the word PIG. As this happens, you should sound out each of the word parts, gradually decreasing the amount of time between the parts until the word becomes a single sound as it is shown as a single word. To increase or decrease the periods of time between the movements of the letters, modify the value of D at line #90. This value represents the interval between movements of 1/60 of a second. For example, the value of D+7 in the listing causes 7/60 of a second to elapse between each movement of the letters across the screen.

Data Entry

All data are entered in response to program messages. The program expects each word to be entered in two parts with a comma separating them. See the example.

Notes and Things to Check

If the words move too slowly or too fast for your child, you should change the value of D at line #90. Increase this value to slow the movement down, and decrease it to speed the movement up. If you make a mistake in entering a word, you can correct it by requesting "CORRECTIONS" and then choosing the item # of the word to correct.

Modification Procedures

Since this is a single purpose program, it is unlikely that you will want to modify it for other uses. Any desired changes will probably be in the

display area. The word display begins at line #600 in the program. One modification you may want to consider is that of changing the colors of the display. This can be accomplished by adding the appropriate commands prior to the PRINT statement at line #720.

```

10 REM      SAVED AT W-SOUNDS
20 REM THIS PROGRAM PROVIDES A METHOD
30 REM FOR USING THE COMPUTER TO
40 REM PRESENT BEGINNING READING
50 REM SKILLS
60 REM *****
70 M=10
80 DIM F$(M),S$(M)
90 D=7
100 GOSUB 210      :REM WORD ENTRY
110 GOSUB 330      :REM REVIEW WORDS
120 GOSUB 600      :REM DISPLAY WORDS
130 REM *****
140 REM      PROGRAM TERMINATION POINT
150 REM *****
160 PRINT
170 PRINT "PROCESSING COMPLETE"
180 PRINT
190 STOP
200 REM *****
210 REM      WORD ENTRY
220 REM *****
230 PRINT "ENTER YOUR WORDS NOW "
240 PRINT "SEPERATE THE SOUNDS WITH A COMMA"
250 PRINT "EXAMPLE P,IG FOR PIG"
260 PRINT
270 I=I+1
280 INPUT F$(I),S$(I)
290 IF F$(I)<>" " THEN 270
300 N=I-1
310 RETURN
320 REM *****
330 REM      REVIEW WORDS
340 REM *****
350 PRINT
360 PRINT "***** WORD REVIEW *****"
370 PRINT "PRESS THE RETURN TO BEGIN";
380 INPUT A$
390 PRINT CHR$(147)
400 PRINT "ITEM #      WORD"
410 FOR I=1 TO N
420 PRINT I;"..";TAB(8);F$(I);"-";S$(I)
430 NEXT I
440 PRINT
450 PRINT "ARE THERE ANY CORRECTIONS (Y OR N)"
460 INPUT A$
470 IF A$<>"Y" THEN 580
480 PRINT "ENTER THE ITEM NUMBER TO CORRECT"
490 INPUT A

```

```

500 IF A<1 OR A>N THEN 560
510 PRINT
520 PRINT "CURRENT VALUE IS ";F$(A);", ";S$(A)
530 PRINT "INPUT CORRECT VALUES";
540 INPUT F$(A),S$(A)
550 GOTO 440
560 PRINT "INVALID ITEM NUMBER"
570 GOTO 450
580 RETURN
590 REM *****
600 REM          DISPLAY WORDS
610 REM *****
620 PRINT
630 PRINT "**** DISPLAY OF WORDS ****"
640 PRINT "PRESS THE RETURN TO BEGIN"
650 INPUT A$
660 PRINT CHR$(147)
670 E=23
680 FOR I=1 TO N
690 S=LEN(F$(I))
700 FOR J=1 TO E-S+1
710 T1=30-J
720 PRINT TAB(5);" ";F$(I);TAB(T1);S$(I);" ";
730 T0=TI
740 IF TI>T0+D THEN 770
750 PRINT CHR$(145)
760 GOTO 740
770 NEXT J
780 PRINT SPC(40);
790 PRINT CHR$(145)
800 NEXT I
810 RETURN

```

```

RUN
ENTER YOUR WORDS NOW
SEPERATE THE SOUNDS WITH A COMMA
EXAMPLE F,IG FOR FIG

```

```

? F,IG
? FF,IG
? TW,IG
? D,IG
?

```

```

***** WORD REVIEW *****
PRESS THE RETURN TO BEGIN?

```

```

ITEM #    WORD
1 ..    F-IG
2 ..    FF-IG
3 ..    TW-IG
4 ..    D-IG

```


ARE THERE ANY CORRECTIONS (Y OR N)
? Y

ENTER THE ITEM NUMBER TO CORRECT
? 2

CURRENT VALUE IS FF,IG
INPUT CORRECT VALUES? F,IG

ARE THERE ANY CORRECTIONS (Y OR N)
? N

**** DISPLAY OF WORDS ****
PRESS THE RETURN TO BEGIN
?

MAJOR SYMBOL TABLE — W-SOUNDS

NAME	DESCRIPTION
A	ITEM NUMBER TO CORRECT
A\$	TEMP ANSWER VARIABLE
D	PRINT DELAY IN 1/60 SECOND
E	END OF PRINT POINTER
F\$	FIRST SOUND
I	INDEX POINTER
J	INDEX POINTER
K	INDEX POINTER
N	NUMBER OF WORDS TO PRINT
S	START OF PRINT POINTER
S\$	SECOND SOUND
TO	SAVED TIME FROM TI FUNCTION
T1	TAB TO LOCATION

FUNCTIONS USED

NAME
CHR\$
GOSUB
LEN
RETURN
SPC
TAB
TI

OUTLINE

Description

This program accepts information from the keyboard and then arranges it into a standard outline format. It can be used for a wide variety of school or home activities, but it will be useful primarily in high school homework and study projects.

Instructions for Use

RUN the program and input the data at the appropriate outline levels of indentation. After you make each entry, the program requests data at the next lower (subordinate) level. If you enter only a carriage return, the program moves back to the next higher level. See the sample RUN. Once you have entered the information, you can modify the entries, and/or store them in a file for later use.

Data Entry

All data are entered during the program's operation.

Notes and Things to Check

Any entry except a RETURN (only), will initiate an entry one level of the outline lower than the preceding entry. If you requested a modification of the outline, the program will print each of the entries in sequence. During this modification, a carriage RETURN causes a line to remain unchanged while an entry of C causes the program to request new information. Note that line 400 of the program provides the lowest level of the outline. In the listing, this is shown as lower-case letters. Although this is not available for normal Commodore 64 screen functions, it is provided in case you are using other devices for output. During the entry of the program, however, you should replace these lower case characters either with other characters or with spaces. In the sample RUN they have been replaced with spaces.

Modification Procedures

Because of the indentation and character manipulation this program has to perform, it may be difficult to modify. In general, each of the major processes of the program has been provided as a subroutine to keep it independent from the others. Be sure you understand the subroutines thoroughly before you make any changes to the program. A remark

statement has been included at line #1420 that you may want to replace with a PRINT I1\$(I) statement in order to analyze the effect of program modifications.

```

10 REM          SAVED AT OUTLINE
20 REM THIS PROGRAM PRODUCES AN OUTLINE
30 REM OF SELECTED ITEMS IN STANDARD
40 REM OUTLINE FORM
50 REM *****
60 M0=200
70 T1=4
80 DIM I$(M0),N$(M0),I1$(M0),A1$(9),A2$(9),A3$(9),A4$(9)
90 GOSUB 350      :REM DATA INITIALIZE
100 PRINT "IS INPUT TO COME FROM A FILE (Y OR N)"
110 INPUT A$
120 IF A$="Y" THEN GOSUB 1460                :REM FILE INPUT
130 IF A$<>"Y" THEN GOSUB 470                :REM OUTLINE INPUT
140 PRINT "SHALL I PRINT THE OUTLINE (Y OR N)"
150 INPUT A$
160 IF A$="Y" THEN GOSUB 840:REM PRINT
170 PRINT "MODIFY THE OUTLINE (Y OR N)"
180 A$=""
190 INPUT A$
200 IF A$="Y" THEN GOSUB 970:REM MODIFY
210 IF A$="Y" THEN 140
220 PRINT "STORE THE OUTLINE (Y OR N)"
230 A$=""
240 INPUT A$
250 IF A$="Y" THEN GOSUB 1180:REM SAVE
260 REM *****
270 REM   PROGRAM TERMINATION POINT
280 REM *****
290 PRINT
300 PRINT
310 PRINT "PROCESSING COMPLETE"
320 PRINT
330 STOP
340 REM *****
350 REM          DATA INITIALIZE
360 REM *****
370 DATA I,II,III,IV,V,VI,VII,VIII,IX
380 DATA A,B,C,D,E,F,G,H,I
390 DATA 1,2,3,4,5,6,7,8,9
400 DATA ,,,,,,
410 READ A1$(1),A1$(2),A1$(3),A1$(4),A1$(5),A1$(6),A1$(7),A1$(8),A1$(9)
420 READ A2$(1),A2$(2),A2$(3),A2$(4),A2$(5),A2$(6),A2$(7),A2$(8),A2$(9)
430 READ A3$(1),A3$(2),A3$(3),A3$(4),A3$(5),A3$(6),A3$(7),A3$(8),A3$(9)
440 READ A4$(1),A4$(2),A4$(3),A4$(4),A4$(5),A4$(6),A4$(7),A4$(8),A4$(9)
450 RETURN
460 REM *****

```

```

470 REM          OUTLINE INPUT
480 REM *****
490 I=1
500 FOR I1=1 TO 9
510 N$(I)="
520 PRINT A1$(I1);
530 INPUT N$(I)
540 IF N$(I)=" THEN 810
550 GOSUB 1290:REM NUMBER CREATE
560 FOR I2=1 TO 9
570 N$(I)="
580 PRINT TAB(T1);A2$(I2);
590 INPUT N$(I)
600 IF N$(I)=" THEN 790
610 GOSUB 1290:REM NUMBER CREATE
620 FOR I3=1 TO 9
630 N$(I)="
640 PRINT TAB(T1*2);A3$(I3);
650 INPUT N$(I)
660 IF N$(I)=" THEN 770
670 GOSUB 1290:REM NUMBER CREATE
680 FOR I4=1 TO 9
690 N$(I)="
700 PRINT TAB(T1*3);A4$(I4);
710 INPUT N$(I)
720 IF N$(I)=" THEN 750
730 GOSUB 1290:REM NBR CREATE
740 NEXT I4
750 I4=0
760 NEXT I3
770 I3=0
780 NEXT I2
790 I2=0
800 NEXT I1
810 M=I-1
820 RETURN
830 REM *****
840 REM          PRINT OUTLINE
850 REM *****
860 PRINT "SHALL I DOUBLE SPACE (Y OR N)";
870 INPUT A$
880 FOR I=1 TO M
890 L=LEN(I$(I))
900 T0=INT(L/2)
910 T=T0*T1+5-LEN(I1$(I))
920 PRINT TAB(T);I1$(I);". ";N$(I)
930 IF A$="Y" THEN PRINT
940 NEXT I
950 RETURN
960 REM *****
970 REM          MODIFY OUTLINE
980 REM *****
990 PRINT
1000 PRINT "EACH OUTLINE ITEM WILL BE PRINTED"
1010 PRINT "THE PROGRAM WILL THEN AWAIT YOUR REPLY"
1020 PRINT "A RETURN WILL MOVE TO THE NEXT ITEM"
1030 PRINT "A C WILL ALLOW YOU TO CHANGE THE DATA"

```

```

1040 PRINT
1050 FOR I=1 TO M
1060 L=LEN(I$(I))
1070 T0=INT(L/2)
1080 T=T0*T1+5-LEN(I1$(I))
1090 PRINT TAB(T);I1$(I);". ";N$(I);
1100 C$=""
1110 INPUT C$
1120 IF C$<>"C" THEN 1150
1130 PRINT "ENTER CORRECT NARRATIVE"
1140 INPUT N$(I)
1150 NEXT I
1160 RETURN
1170 REM *****
1180 REM             SAVE OUTLINE
1190 REM *****
1200 PRINT "ENTER FILE NAME TO WRITE THE DATA TO"
1210 INPUT F$
1220 OPEN 1,1,1,F$
1230 FOR I=1 TO M
1240 PRINT#1,I$(I) ", " I1$(I) ", " N$(I)
1250 NEXT I
1260 CLOSE 1
1270 RETURN
1280 REM *****
1290 REM             NUMBER CREATE
1300 REM *****
1310 I$(I)=RIGHT$(STR$(I1),1)
1320 IF I2=0 THEN 1380
1330 I$(I)=I$(I)+". "+RIGHT$(STR$(I2),1)
1340 IF I3=0 THEN 1380
1350 I$(I)=I$(I)+". "+RIGHT$(STR$(I3),1)
1360 IF I4=0 THEN 1380
1370 I$(I)=I$(I)+". "+RIGHT$(STR$(I4),1)
1380 IF I1<>0 THEN I1$(I)=A1$(I1)
1390 IF I2<>0 THEN I1$(I)=A2$(I2)
1400 IF I3<>0 THEN I1$(I)=A3$(I3)
1410 IF I4<>0 THEN I1$(I)=A4$(I4)
1420 REM
1430 I=I+1
1440 RETURN
1450 REM *****
1460 REM             FILE INPUT
1470 REM *****
1480 PRINT "ENTER THE NAME OF THE INPUT FILE"
1490 INPUT F$
1500 OPEN 1,1,0,F$
1510 I=1
1520 IF ST<>0 THEN 1560
1530 INPUT#1,I$(I),I1$(I),N$(I)
1540 I=I+1
1550 GOTO 1520
1560 CLOSE 1
1570 M=I-1
1580 RETURN

```

RUN
IS INPUT TO COME FROM A FILE (Y OR N)
? N
I? PRESIDENTS
 A? NAMES
 1? YEARS
 ?
 2? RECORDS
 ?
 3?
 B? POLITICAL NOMINATION PROCESS
 1?
 C?
II? CABINETS
 A?
III?
SHALL I PRINT THE OUTLINE (Y OR N)
? Y
SHALL I DOUBLE SPACE (Y OR N)? N

I. PRESIDENTS
 A. NAMES
 1. YEARS
 2. RECORDS
 B. POLITICAL NOMINATION PROCESS
II. CABINETS
MODIFY THE OUTLINE (Y OR N)
? Y

EACH OUTLINE ITEM WILL BE PRINTED
THE PROGRAM WILL THEN AWAIT YOUR REPLY
A RETURN WILL MOVE TO THE NEXT ITEM
A C WILL ALLOW YOU TO CHANGE THE DATA

I. PRESIDENTS?
 A. NAMES?
 1. YEARS?
 2. RECORDS? C
ENTER CORRECT NARRATIVE
? POLITICAL RECORD
 B. POLITICAL NOMINATION PROCESS?
II. CABINETS?
SHALL I PRINT THE OUTLINE (Y OR N)
? Y

SHALL I DOUBLE SPACE (Y OR N)? N
I. PRESIDENTS
 A. NAMES
 1. YEARS
 2. POLITICAL RECORD

B. POLITICAL NOMINATION PROCESS
II. CABINETS
MODIFY THE OUTLINE (Y OR N)
? N
STORE THE OUTLINE (Y OR N)
? Y
ENTER FILE NAME TO WRITE THE DATA TO
?OUTLINE

PRESS RECORD & PLAY ON TAPE
OK

PROCESSING COMPLETE

Break in 330

RUN
IS INPUT TO COME FROM A FILE (Y OR N)
? Y
ENTER THE NAME OF THE INPUT FILE
? OUTLINE-1
PRESS PLAY & RECORD ON TAPE
OK
SHALL I PRINT THE OUTLINE (Y OR N)
? Y
SHALL I DOUBLE SPACE (Y OR N)? N
I. PRESIDENTS
A. NAMES
1. YEARS
2. POLITICAL RECORDS
B. POLITICAL NOMINATION PROCESS
II. CABINETS
MODIFY THE OUTLINE (Y OR N)
? N
STORE THE OUTLINE (Y OR N)
? N

PROCESSING COMPLETE

Break in 330

MAJOR SYMBOL TABLE — OUTLINE

NAME	DESCRIPTION
A\$	TEMP ANSWER VARIABLE
A1\$	OUTLINE POSITION VALUE
A2\$	OUTLINE POSITION VALUE
A3\$	OUTLINE POSITION VALUE
A4\$	OUTLINE POSITION VALUE
C\$	CHANGE REQUIRED VARIABLE
F\$	FILE NAME
I	INDEX AND ARRAY POINTER
I\$()	INDEX # ARRAY
I1	INDEX TO OUTLINE LEVEL
I1\$()	ARRAY FOR OUTLINE POSITIONING
I2	INDEX TO OUTLINE LEVEL
I3	INDEX TO OUTLINE LEVEL
I4	INDEX TO OUTLINE LEVEL
J	INDEX AND ARRAY POINTER
L	LEN OF DATA
M	NUMBER OF ITEMS
MO	MAXIMUM NUMBER OF ITEMS
N\$()	NARRATIVE ARRAY
T	TAB AMOUNT
TO	COMPUTED TAB POSITION
T1	SPACING FOR INSET

FUNCTIONS USED

NAME
CHR\$
CLOSE
DIM
GOSUB
LEN
OPEN
RETURN
RIGHT\$
TAB

QA-BUILD

Description

This program works in conjunction with QA-TUTOR to prepare and present review questions for reviewing homework and studying for tests. Teachers can also use it to prepare self-test materials for their students.

Instructions for Use

RUN the program and respond to the messages from the program. The LESSON-NAME entry will be used as the name of the storage file for later retrieval from your tape (or disk). After you have entered all the questions and their answers, you are given the opportunity to review and/or change the data that have been entered.

Data Entry

All data are entered in response to program messages.

Notes and Things to Check

Make sure that your tape is positioned properly before you RUN the program. Data will continue to be requested until you give a carriage RETURN only in response to the program's request for the next question. If you press another key along with RETURN, you will be obliged to use the correction process to change it.

Modification Procedures

Since the output of this program is intended primarily as a file for use with QA-TUTOR, it is necessary to review the file output to determine the ultimate effect of any changes that you may make. You can do this by using either the QA-TUTOR program or the PRINT-TF program provided in Chapter 9. Note that any modifications to the "GET INPUT" procedure (Lines 670-750) will also apply to the correction process since this routine is entered whenever corrections are requested.

```
10 REM    SAVED AT QA-BUILD
20 REM    TUTOR PROGRAM FOR QUESTION AND
30 REM    ANSWER REVIEW
40 X$="*****"
50 M=100
60 PRINT "ENTER THE NAME OF THE OUTPUT FILE"
```

```

70 INPUT F1$
80 OPEN 1,1,1,F1$
90 DIM Q$(M),A0$(M)
100 PRINT "ENTER THE DATA NOW (END WITH A RETURN)"
110 FOR I=1 TO M
120 Q$(I)=" "
130 GOSUB 670 :REM GET INPUT
140 IF Q$(I)=" " THEN 160
150 NEXT I
160 I1=I-1
170 PRINT "SHALL I PRINT THE DATA (Y OR N)"
180 INPUT A$
190 IF A$="Y" THEN GOSUB 300:REM PRINT
200 GOSUB 460 :REM PRINT TO FILE
210 REM *****
220 REM PROGRAM TERMINATION POINT
230 REM *****
240 PRINT
250 PRINT
260 PRINT "PROCESSING COMPLETE"
270 PRINT
280 STOP
290 REM *****
300 REM PRINT DATA
310 REM *****
320 PRINT
330 PRINT X$
340 PRINT " #";TAB(5);"QUESTION"
350 PRINT TAB(7);"ANSWER"
360 PRINT
370 FOR I=1 TO I1
380 PRINT I;TAB(5);Q$(I)
390 PRINT TAB(7);A0$(I)
400 NEXT I
410 PRINT "WOULD YOU LIKE TO CORRECT THE DATA (Y OR N)"
420 INPUT A1$
430 IF A1$="Y" THEN GOSUB 540:REM CORRECT DATA
440 RETURN
450 REM *****
460 REM PRINT TO FILE
470 REM *****
480 FOR I= 1 TO I1
490 PRINT#1,Q$(I) ", " A0$(I)
500 NEXT I
510 CLOSE 1
520 RETURN
530 REM *****
540 REM CORRECTION ROUTINE
550 REM *****
560 PRINT "ENTER THE ITEM NUMBER TO CHANGE";
570 INPUT C
580 I=C
590 PRINT "ENTER THE CORRECT DATA"
600 GOSUB 670 :REM GET INPUT
610 PRINT "ANY MORE CHANGES (Y OR N)";
620 A1$=" "
630 INPUT A1$

```

```

640 IF A1$="Y" THEN 560
650 RETURN
660 REM *****
670 REM             GET INPUT
680 REM *****
690 PRINT "QUESTION";I;
700 Q$(I)="":INPUT Q$(I)
710 IF Q$(I)=" " THEN 750
720 PRINT "ANSWER";I;
730 INPUT A0$(I)
740 PRINT
750 RETURN

```

```

RUN
ENTER THE NAME OF THE OUTPUT FILE
? LESSON-1
PRESS RECORD & PLAY ON TAPE
OK
ENTER THE DATA NOW (END WITH A RETURN)
QUESTION 1 ? WHO WAS THE FIRST PRESIDENT

ANSWER 1 ? GEORGE WASHINGTON

QUESTION 2 ? WHO WAS THE SECOND PRESIDEN
T
ANSWER 2 ? JOHN ADAMS

QUESTION 3 ? WHO WAS THE THIRD PRESIDENT

ANSWER 3 ? BEN FRANKLIN

QUESTION 4 ?
SHALL I PRINT THE DATA (Y OR N)
? Y

```

```

*****
# QUESTION
  ANSWER

1 WHO WAS THE FIRST PRESIDENT
  GEORGE WASHINGTON
2 WHO WAS THE SECOND PRESIDENT
  JOHN ADAMS
3 WHO WAS THE THIRD PRESIDENT
  BEN FRANKLIN
WOULD YOU LIKE TO CORRECT THE DATA (Y OR
N)
? Y
ENTER THE ITEM NUMBER TO CHANGE? 3

```

ENTER THE CORRECT DATA
QUESTION 3 ? WHO WAS THE THIRD PRESIDENT

ANSWER 3 ? THOMAS JEFFERSON

ANY MORE CHANGES (Y OR N)? N

PROCESSING COMPLETE

Break in 280

MAJOR SYMBOL TABLE — QA-BUILD

NAME	DESCRIPTION
A0\$()	ANSWER ARRAY
A\$	PRINT? ANSWER
A1\$	CORRECTIONS? ANSWER
C	ITEM # TO CORRECT
F1\$	FILE NAME
I	INDEX AND ARRAY POINTER
I1	NUMBER OF RECORDS READ
M	MAXIMUM NUMBER OF RECORDS ALLOWED
Q\$()	QUESTION ARRAY
X\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
CLOSE
DIM
GOSUB
OPEN
RETURN

QA-TUTOR

Description

This program presents a question-and-answer review session for students, using files created by the program QA-BUILD. It can be used to assist in test study, homework review, or self-testing on specific subject areas.

Instructions for Use

RUN the program and answer its request for a file name with a file that has been created by the QA-BUILD program. When you indicate that you are ready for the question-and-answer session, the program asks each question and then waits for your answer. It then checks that answer against the answer in the file. If the answer is incorrect, it gives you a second try. If your second answer is also incorrect, it gives you the correct answer and moves on to the next question. After you answer all the questions, the program presents your overall accuracy in summary form.

Data Entry

All data are provided from a file created by the QA-BUILD program.

Notes and Things to Check

If this program fails to operate correctly, the source of the difficulty is probably somewhere in the data file created by the QA-BUILD program. That file can be checked through the use of the PRINT-TF program that appears in Chapter 9. If the file is stored on tape, you may find that it cannot be located because the tape was not positioned properly prior to either program's operation.

Modification Procedures

For ease of modification, the program is divided into separate routines. You are most likely to want to modify the area of the program where the questions are presented (lines 520-730). Note that the display of results results from this subroutine after all the questions have been asked. If you would like to remove the capability of displaying the answers prior to the question and answer session, delete the statements at lines 100, 120, and 130.

```

10 REM      SAVED AT QA-TUTOR
20 REM      QUESTION AND ANSWER TUTORING
30 X$="*****"
40 M=100
50 PRINT "ENTER THE NAME OF THE INPUT FILE"
60 INPUT F1$
70 OPEN 1,1,0,F1$
80 DIM Q$(M),A0$(M)
90 GOSUB 430      :REM GET INPUT
100 I1=I-1
110 PRINT "SHALL I PRINT THE DATA (Y OR N)";
120 INPUT A$
130 IF A$="Y" THEN GOSUB 270:REM PRINT
140 IF A$<>"Y" THEN PRINT I1;"RECORDS WERE FOUND"
150 PRINT "ARE YOU READY FOR A QUESTION/ANSWER SESSION (Y OR N) "
160 INPUT A$
170 IF A$="Y" THEN GOSUB 520:REM Q-A
180 REM *****
190 REM PROGRAM TERMINATION POINT
200 REM *****
210 PRINT
220 PRINT
230 PRINT "PROCESSING COMPLETE"
240 PRINT
250 STOP
260 REM *****
270 REM      PRINT DATA
280 REM *****
290 PRINT
300 PRINT X$
310 PRINT " #";TAB(5);">QUESTION<"
320 PRINT TAB(7);">ANSWER<"
330 PRINT
340 FOR I=1 TO I1
350 PRINT I;TAB(5);
360 PRINT ">";Q$(I);"<"
370 PRINT TAB(7);">";A0$(I);"<"
380 PRINT
390 NEXT I
400 PRINT "EOF"
410 RETURN
420 REM *****
430 REM      GET INPUT
440 REM *****
450 FOR I=1 TO M
460 IF ST<>0 THEN 490
470 INPUT#1,Q$(I),A0$(I)
480 NEXT I
490 CLOSE 1
500 RETURN
510 REM *****
520 REM      Q-A AREA
530 REM *****
540 C=0
550 W=0
560 FOR I=1 TO I1
570 PRINT

```

```

580 PRINT Q$(I)
590 INPUT A9$
600 IF A9$=A0$(I) THEN 690
610 W=W+1
620 PRINT "THAT'S ";W
630 PRINT "BAD ANSWER TRY AGAIN"
640 INPUT A9$
650 IF A9$=A0$(I) THEN 700
660 PRINT "STILL INCORRECT - THE CORRECT ANSWER IS"
670 PRINT A0$(I)
680 GOTO 710
690 C=C+1
700 PRINT "CORRECT"
710 NEXT I
720 GOSUB 750 :REM DISPLAY RESULTS
730 RETURN
740 REM *****
750 REM          DISPLAY RESULTS
760 REM *****
770 PRINT "*****"
780 PRINT "          SCORE BOARD"
790 PRINT
800 PRINT "NUMBER CORRECT ....";C
810 PRINT "NUMBER INCORRECT ..";W
820 PRINT
830 PRINT "PERCENTAGE SCORE ..";C/(C+W)*100
840 RETURN

```

```

RUN
ENTER THE NAME OF THE INPUT FILE
? LESSON-1
PRESS PLAY ON TAPE
OK
SHALL I PRINT THE DATA (Y OR N)? Y

```

```
*****
```

```

# >QUESTION<
  >ANSWER<

1. >WHO WAS THE FIRST PRESIDENT<
   >GEORGE WASHINGTON<

2. >WHO WAS THE SECOND PRESIDENT<
   >JOHN ADAMS<

3. >WHO WAS THE THIRD PRESIDENT<
   >THOMAS JEFFERSON<

```

```

EOF
ARE YOU READY FOR A QUESTION/ANSWER SESSION (Y OR N)
? Y

```

WHO WAS THE FIRST PRESIDENT
? GEORGE WASHINGTON
CORRECT

WHO WAS THE SECOND PRESIDENT
? JOE SMITH
THAT'S 1
BAD ANSWER TRY AGAIN
? BEN FRANKLIN
STILL INCORRECT - THE CORRECT ANSWER IS
JOHN ADAMS

WHO WAS THE THIRD PRESIDENT
? THOMAS JEFFERSON

CORRECT

SCORE BOARD

NUMBER CORRECT 2
NUMBER INCORRECT .. 1

PERCENTAGE SCORE .. 66.66667

PROCESSING COMPLETE

Break in 250

MAJOR SYMBOL TABLE — QA-TUTOR

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
AO\$()	ANSWER ARRAY
A9\$	ANSWER TRIED
C	NUMBER CORRECT ANSWERS
F1\$	FILE NAME
I	INDEX AND ARRAY POINTER
I1	NUMBER OF RECORDS READ
M	MAXIMUM NUMBER OF RECORDS ALLOWED
Q\$()	QUESTION ARRAY
W	NUMBER WRONG
X\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
CLOSE
DIM
GOSUB
OPEN
RETURN
ST

8 Outside Activities and Recreation

TRIPPLAN

Description

This program provides assistance in planning trips and vacations and estimating their total costs. It can be used to evaluate alternative plans and estimate financial needs during a trip.

Instructions for Use

RUN the program and respond to the program's messages. To simplify processing and the format of the output, you should enter expenses in whole dollars only. If you are using the screen to display results and the number of days exceeds two, you must press the CTRL key to slow the movement of the output.

Data Entry

All data are entered in response to the program's requests.

Notes and Things to Check

Make sure that the entry for each item requested does not contain any alphabetic characters if the program is looking only for numbers. For example, to enter 50 dollars as a cost, you should enter it as 50, not as fifty or \$50.

Modification Procedures

All the program's processes are performed in the loop at lines 270-430. The preceding lines accept the data input and the following lines

print the summary. If you are using the screen for output and your trips are consistently longer than two days, you may want to add an appropriate statement every time the variable J is an even number. This will delay processing until you are ready to view the next two days. It can be accomplished by requesting input every second day; the program will then await your RETURN before it displays the next day.

```

10 REM          SAVED AT TRIPPLAN
20 REM THIS PROGRAM ASSISTS IN TRIP
25 REM TRIP PLANNING AND COST ESTIMATES
30 REM *****
40 M0=15
50 DIM D$(M0),M(M0),F(M0),L(M0),O(M0),T(M0)
60 INPUT "ENTER TRIP NAME ";N$
70 INPUT "ENTER NUMBER OF DAYS";D1
80 INPUT "ENTER EXPECTED MPG";C
90 INPUT "ENTER AVG COST/GAL FUEL";G
100 FOR J=1 TO D1
110 PRINT "FOR DAY";J;"ENTER"
120 INPUT "    MILES TO TRAVEL";M(J)
130 INPUT "    STOPPING POINT";D$(J)
140 INPUT "    COST FOR MEALS";F(J)
150 INPUT "    COST FOR LODGING";L(J)
160 INPUT "    OTHER COSTS";O(J)
170 NEXT J
180 REM *****
190 REM          PRINT RESULTS
200 REM *****
210 X$="*****"
220 PRINT X$
230 PRINT "    TRIP ANALYSIS"
240 PRINT X$
250 PRINT "TRIP NAME: ";N$
260 PRINT
270 FOR J=1 TO D1
280 X=M(J)/C*G
290 PRINT "DAY";J;TAB(8);"TRAVEL ";M(J);"MILES TO ";D$(J)
300 PRINT "GAS";TAB(8);"FOOD";TAB(16);"LODGING";TAB(24);
310 PRINT "OTHER";TAB(32);"TOTAL"
320 PRINT TAB(1);X;TAB(8);F(J);TAB(16);L(J);TAB(24);O(J);TAB(32);
330 T(J)=X+F(J)+L(J)+O(J)
340 PRINT TAB(32);T(J)
350 X1=X1+X
360 M1=M1+M(J)
370 F1=F1+F(J)
380 L1=L1+L(J)
390 O1=O1+O(J)
400 T1=T1+T(J)
410 T(J)=0
420 PRINT
430 NEXT J

```

```

440 PRINT "TOTALS";TAB(15);"MILES TRAVELED";M1
450 PRINT "GAS";TAB(8);"FOOD";TAB(16);"LODGING";TAB(24);"OTHER";TAB(32
);"TOTAL"
460 PRINT TAB(1);X1;TAB(8);F1;TAB(16);L1;TAB(24);O1;TAB(32);T1
470 REM *****
480 REM PROGRAM TERMINATION POINT
490 REM *****
500 PRINT
510 PRINT "PROCESSING COMPLETE"
520 PRINT
530 STOP

```

```

RUN
ENTER TRIP NAME ? DENVER VACATION
ENTER NUMBER OF DAYS? 2
ENTER EXPECTED MPG? 15
ENTER AVG COST/GAL FUEL? 1.25
FOR DAY 1 ENTER
MILES TO TRAVEL? 300
STOPPING POINT? CHEYENNE
COST FOR MEALS? 75
COST FOR LODGING? 100
OTHER COSTS? 50
FOR DAY 2 ENTER
MILES TO TRAVEL? 150
STOPPING POINT? DENVER
COST FOR MEALS? 75
COST FOR LODGING? 0
OTHER COSTS? 50

```

TRIP ANALYSIS

TRIP NAME: DENVER VACATION

DAY 1	TRAVEL	300 MILES TO CHEYENNE		
GAS	FOOD	LODGING	OTHER	TOTAL
25	75	100	50	250
DAY 2	TRAVEL	150 MILES TO DENVER		
GAS	FOOD	LODGING	OTHER	TOTAL
12.5	75	0	50	137.5

TOTALS			MILES TRAVELED	450	
GAS	FOOD		LODGING	OTHER	TOTAL
37.5	150		100	100	387.5

PROCESSING COMPLETE

Break in 530

MAJOR SYMBOL TABLE — TRIPPLAN

NAME	DESCRIPTION
C	FUEL CONSUMPTION
D\$()	DESTINATION ARRAY
D1	NUMBER OF DAYS
F()	DAILY FOOD COST ARRAY
F1	TOTAL FOOD COSTS
G	COST PER GALLON
J	INDEX AND ARRAY POINTER
L()	DAILY LODGING COST ARRAY
L1	TOTAL LODGING COSTS
M()	DAILY MILES ARRAY
MO	MAXIMUM NUMBER OF DAYS
N\$	TRIP NAME
O()	OTHER COSTS ARRAY
O1	TOTAL OTHER COSTS
T	TOTAL COSTS
T()	DAILY TOTAL COST ARRAY
X	FUEL COST
X\$	LINE OF ASTERISKS
X1	TOTAL FOOD COSTS

FUNCTIONS USED

NAME
DIM
TAB

DICEROLL

Description

This program rolls dice on the screen using the images of actual dice. It can be used for game playing in the same way regular dice are used. The dice values are computed and are randomly chosen for each roll.

Instructions for Use

RUN the program and tell it the number of dice to roll. If you respond with a zero or just with a RETURN, the program will end.

Data Entry

Not applicable to this program.

Notes and Things to Check

The program will display no more than five dice at a time because of the limitation of screen size.

Modification Procedures

The dice rolls are determined in the loop at lines 100–120 of the program, and the results are displayed by the "PRINT DICE" subroutine. If you would like to clear the screen after each roll you can add this statement:

```
95 PRINT CHR$(147)
```

```
10 REM          SAVED AT DICEROLL
20 REM  THIS PROGRAM ROLLS UP TO 5 DICE
25 REM  TO THE SCREEN
30 REM  *****
40 DIM D(5),L$(3,6)
50 Y=0
60 GOSUB 370      :REM SET DICE IMAGES
70 D1=0
80 INPUT "ENTER THE NUMBER OF DICE TO ROLL";D1
90 IF D1=0 THEN 150
100 FOR R1=1 TO D1
110 D(R1)=INT((RND(Y)*6+1))
120 NEXT R1
130 GOSUB 180     :REM PRINT DICE
140 GOTO 70
150 REM ** PROGRAM TERMINATION POINT **
160 PRINT
```

```

170 STOP
180 REM *****
190 REM PRINT DICE
200 REM *****
210 FOR I=1 TO D1
220 PRINT "----- ";
230 NEXT I
240 PRINT
250 FOR J=1 TO 3
260 FOR I=1 TO D1
270 PRINT "I";L$(J,D(I));"I ";
280 NEXT I
290 PRINT
300 NEXT J
310 FOR I=1 TO D1
320 PRINT "----- ";
330 NEXT I
340 PRINT
350 RETURN
360 REM *****
370 REM SET DICE IMAGES
380 REM *****
390 L$(1,1)=" "
400 L$(2,1)=" * "
410 L$(3,1)=" "
420 L$(1,2)=" * "
430 L$(2,2)=" "
440 L$(3,2)=" * "
450 L$(1,3)=" * "
460 L$(2,3)=" * "
470 L$(3,3)=" * "
480 L$(1,4)=" * * "
490 L$(2,4)=" "
500 L$(3,4)=" * * "
510 L$(1,5)=" * * "
520 L$(2,5)=" * "
530 L$(3,5)=" * * "
540 L$(1,6)=" * * "
550 L$(2,6)=" * * "
560 L$(3,6)=" * * "
570 RETURN

```

RUN

ENTER THE NUMBER OF DICE TO ROLL? 4

```

-----
I * * I I * I I * * I I * * I
I * I I I I I I * * I
I * * I I * I I * * I I * * I
-----

```

ENTER THE NUMBER OF DICE TO ROLL? 3

```

-----
I * * I I * * I I I
I * I I * * I I * I
I * * I I * * I I I
-----

```

ENTER THE NUMBER OF DICE TO ROLL? 2

```
-----  
I * * I I      I  
I * * I I * I  
I * * I I      I  
-----
```

ENTER THE NUMBER OF DICE TO ROLL?

Break in 170

MAJOR SYMBOL TABLE — DICEROLL

NAME	DESCRIPTION
D()	NUMBER ROLLED ARRAY
D1	# OF DICE TO ROLL
I	INDEX AND ARRAY POINTER
J	INDEX AND ARRAY POINTER
L()	DICE LINE IMAGE ARRAY
R1	INDEX FOR ROLL NUMBER
Y	SEED FOR RND

FUNCTIONS USED

NAME
DIM
GOSUB
INT
RETURN

FOOTBALL

Description

This program is designed to provide football fans with a method of predicting game outcomes. It uses a variety of factors to rate the two teams and ultimately to pick the winner.

Instructions for Use

The teams are compared by various factors, which need to be named in the first DATA statements of the program. Following the factors, the names of the individual teams must be entered. The individual teams must then be compared on the basis of the ratings for each of the factors specified in the earlier statement. These statements are shown in the program listing at lines 1350-1410. The program can then be run throughout the season to help you pick the winners.

During each RUN of the program, you can specify those additional factors that may influence a game's result on a given weekend. These factors are entered in response to the program's messages. The program will then provide its calculated outcome based upon the ratings that you have given the two teams. Because individual factors may change throughout the season, you will have to change the DATA statements to reflect changed ratings as each team's current performance level changes.

Data Entry

Data are provided in two forms:

1. DATA statements are used to record data that remain relatively unchanged from week to week. These data are entered in three sections. The first contains information to specify the factors used to rate the teams. This entry is provided in the following form:

DATA # of factors, name of factor 1, name of factor 2, and so on.

The next section contains the same information for each team. It is provided in the following form:

DATA # of teams, name of team 1, name of team 2, and so on.

The last section contains individual ratings for each team. It is in the following form:

DATA team name, factor 1 rating, factor 2 rating, and so on.

One numeric rating is required for each factor specified. This rating is normally in the range of 1 to 5.

2. Game-specific factors are entered at the program's request.

Notes and Things to Check

Game prediction is at best a nebulous science. It may take you several attempts at choosing factors and ratings before you are satisfied with the results. You should consider this program as a framework within which to develop your own prediction scheme rather than a final predictor in its present form. Be sure you have the required number of entries for each team in the DATA statements and that each is separated from the others by a comma.

Modification Procedures

Because this program was designed to be modified substantially by any user, each of its functions is carefully insulated from others. You can easily modify the input or calculation processes without causing difficulty to other program sections. For example, you can add more week-to-week factors by including them in the "ENTER OTHER FACTORS" routine. You can also modify the calculation of each factor's result by noting the comments in the "CALCULATION AREA."

```
10 REM          SAVED AT FOOTBALL
20 REM THIS PROGRAM PROVIDES A METHOD
30 REM THAT COULD BE USED TO HELP
40 REM PICK FOOTBALL GAME WINNERS
50 REM *****
60 PRINT "ENTERING THE FACTORS FOR EACH TEAM"
70 READ M
80 DIM F$(M)
90 FOR J=1 TO M
100 READ F$(J)
110 NEXT J
120 PRINT "ENTERING THE TEAM NAMES"
130 READ N
140 DIM T$(N),F(N,M)
150 FOR I=1 TO N
160 READ T$(I)
170 NEXT I
180 PRINT "ENTERING THE FACTORS FOR EACH TEAM"
190 FOR I=1 TO N
200 READ T$(I)
210 FOR J=1 TO M
220 READ F(I,J)
230 NEXT J
240 NEXT I
250 PRINT "BASIC DATA HAS BEEN ENTERED"
260 PRINT
```

```

270 T=0
280 I1=0
290 J1=0
300 PRINT "ENTER THE GAMES TO PROJECT"
310 INPUT "HOME TEAM";H$
320 INPUT "VISITING TEAM";V$
330 FOR I=1 TO N
340 IF H$=T$(I) THEN I1=I
350 IF V$=T$(I) THEN J1=I
360 NEXT I
370 IF I1<>0 THEN 390
380 PRINT "HOME TEAM NOT FOUND"
390 IF J1<>0 THEN 410
400 PRINT "HOME TEAM NOT FOUND"
410 IF I1=0 OR J1=0 THEN 300
420 PRINT
430 GOSUB 640 :REM ENTER OTHER FACTORS
440 PRINT
450 INPUT "SHALL I PRINT DETAILS";A$
460 PRINT CHR$(147)
470 PRINT "*****CALCULATING*****"
480 GOSUB 850 :REM CALCULATION AREA
490 PRINT
500 X$="*****"
510 PRINT X$
520 PRINT H$;"(HOME) VS ";V$
530 PRINT
540 IF T<0 THEN PRINT V$;" BY ";-T
550 IF T>=0 THEN PRINT H$;" BY ";T
560 PRINT X$:INPUT "ANOTHER GAME (Y OR N)";A$
570 IF A$="Y" THEN 270
580 REM *****
590 REM PROGRAM TERMINATION POINT
600 REM *****
610 PRINT
620 STOP
630 REM *****
640 REM ENTER OTHER FACTORS
650 REM *****
660 PRINT "ENTER MORALE FACTOR (1-5)"
670 PRINT H$;
680 INPUT M0(1)
690 PRINT V$;
700 INPUT M0(2)
710 PRINT "ENTER INTENSITY FACTOR 1-5"
720 PRINT H$;
730 INPUT I0(1)
740 PRINT V$;
750 INPUT I0(2)
760 PRINT "ENTER HEALTH FACTOR (1-5)"
770 PRINT H$;
780 INPUT H0(1)
790 PRINT V$;
800 INPUT H0(2)
810 PRINT "ENTER HOME FIELD ADVANTAGE (1-3)";
820 INPUT F0
830 RETURN

```

```

840 REM *****
850 REM          CALCULATION AREA
860 REM    EACH FACTOR IS SEPARATE TO FACILITATE CHANGE
870 REM *****
880 IF A$<>"Y" THEN 920
890 PRINT X$
900 PRINT "FACTOR";TAB(15);H$;TAB(25);V$;TAB(35);"DIFF"
910 PRINT "-----"
920 D=F(I1,1)-F(J1,1) :REM FACTOR 1
930 T=D
940 IF A$="Y" THEN PRINT F$(1);TAB(15);F(I1,1);TAB(25);F(J1,1);TAB(35)
;D
950 D=F(I1,2)-F(J1,2) :REM FACTOR 2
960 T=T+D
970 IF A$="Y" THEN PRINT F$(2);TAB(15);F(I1,2);TAB(25);F(J1,2);TAB(35)
;D
980 D=F(I1,3)-F(J1,3) :REM FACTOR 3
990 T=T+D
1000 IF A$="Y" THEN PRINT F$(3);TAB(15);F(I1,3);TAB(25);F(J1,3);TAB(35)
;D
1010 D=F(I1,4)-F(J1,4) :REM FACTOR 4
1020 T=T+D
1030 IF A$="Y" THEN PRINT F$(4);TAB(15);F(I1,4);TAB(25);F(J1,4);TAB(35)
);D
1040 D=F(I1,5)-F(J1,5) :REM FACTOR 5
1050 T=T+D
1060 IF A$="Y" THEN PRINT F$(5);TAB(15);F(I1,5);TAB(25);F(J1,5);TAB(35)
);D
1070 D=F(I1,6)-F(J1,6) :REM FACTOR 6
1080 T=T+D
1090 IF A$="Y" THEN PRINT F$(6);TAB(15);F(I1,6);TAB(25);F(J1,6);TAB(35)
);D
1100 D=F(I1,7)-F(J1,7) :REM FACTOR 7
1110 T=T+D
1120 IF A$="Y" THEN PRINT F$(7);TAB(15);F(I1,7);TAB(25);F(J1,7);TAB(35)
);D
1130 D=F(I1,8)-F(J1,8) :REM FACTOR 8
1140 T=T+D
1150 IF A$="Y" THEN PRINT F$(8);TAB(15);F(I1,8);TAB(25);F(J1,8);TAB(35)
);D
1160 D=M0(1)-M0(2) :REM MORALE
1170 T=T+D
1180 IF A$="Y" THEN PRINT "MORALE";TAB(15);M0(1);TAB(25);M0(2);TAB(35)
;D
1190 D=I0(1)-I0(2) :REM INTENSITY
1200 T=T+D
1210 IF A$="Y" THEN PRINT "INTENSITY";TAB(15);I0(1);TAB(25);I0(2);TAB(
35);D
1220 D=H0(1)-H0(2) :REM HEALTH
1230 T=T+D
1240 IF A$="Y" THEN PRINT "HEALTH";TAB(15);H0(1);TAB(25);H0(2);TAB(35)
;D
1250 D=F0 :REM HOME TEAM ADVANTAGE
1260 T=T+D
1270 IF A$="Y" THEN PRINT "HOME";TAB(15);F0;TAB(35);D
1280 RETURN
1290 REM *****

```

```
1300 REM          DATA ENTRY FOLLOWS
1310 REM *****
1320 REM NAME OF FACTORS FIRST
1330 DATA 9,RUSH DEFENSE,RUSH OFFENSE,PASS DEFENSE,PASS OFFENSE
1340 DATA SCORING DEF,SCORING OFF,EXP,PUNTING,TURNOVERS
1350 REM NAMES OF TEAMS FOLLOW
1360 DATA 4,TEAM 1,TEAM 2,TEAM 3,TEAM 4
1370 REM FACTORS FOR EACH TEAM FOLLOW (1-5 EACH)
1380 DATA TEAM 1,3,1,1,4,2,2,1,2,3
1390 DATA TEAM 2,3,2,4,2,5,2,2,2,2
1400 DATA TEAM 3,3,4,1,1,2,3,2,2,2
1410 DATA TEAM 4,3,3,4,3,3,3,3,3,1
```

```
RUN
ENTERING THE FACTORS FOR EACH TEAM
ENTERING THE TEAM NAMES
ENTERING THE FACTORS FOR EACH TEAM
BASIC DATA HAS BEEN ENTERED
```

```
ENTER THE GAMES TO PROJECT
HOME TEAM? TEAM 1
VISITING TEAM? TEAM 2
```

```
ENTER MORALE FACTOR (1-5)
TEAM 1? 1
TEAM 2? 2
ENTER INTENSITY FACTOR 1-5
TEAM 1? 2
TEAM 2? 2
ENTER HEALTH FACTOR (1-5)
TEAM 1? 3
TEAM 2? 2
ENTER HOME FIELD ADVANTAGE (1-3)? 2
```

```
SHALL I PRINT DETAILS? N
```

```
*****CALCULATING*****
```

```
*****
TEAM 1(HOME) VS TEAM 2
```

```
TEAM 2 BY 4
*****
ANOTHER GAME (Y OR N)? N
```

```
Break in 620
```

RUN
 ENTERING THE FACTORS FOR EACH TEAM
 ENTERING THE TEAM NAMES
 ENTERING THE FACTORS FOR EACH TEAM
 BASIC DATA HAS BEEN ENTERED

ENTER THE GAMES TO PROJECT
 HOME TEAM? TEAM 1
 VISITING TEAM? TEAM 2

ENTER MORALE FACTOR (1-5)
 TEAM 1? 2
 TEAM 2? 3
 ENTER INTENSITY FACTOR 1-5
 TEAM 1? 4
 TEAM 2? 1
 ENTER HEALTH FACTOR (1-5)
 TEAM 1? 2
 TEAM 2? 3
 ENTER HOME FIELD ADVANTAGE (1-3)? 1

SHALL I PRINT DETAILS? Y

*****CALCULATING*****

FACTOR	TEAM 1	TEAM 2	DIFF
RUSH DEFENSE	3	3	0
RUSH OFFENSE	1	2	-1
PASS DEFENSE	1	4	-3
PASS OFFENSE	4	2	2
SCORING DEF	2	5	-3
SCORING OFF	2	2	0
EXP	1	2	-1
PUNTING	2	2	0
MORALE	2	3	-1
INTENSITY	4	1	3
HEALTH	2	3	-1
HOME	1		1

 TEAM 1(HOME) VS TEAM 2

TEAM 2 BY 4

 ANOTHER GAME (Y OR N)? N

Break in 620

MAJOR SYMBOL TABLE — FOOTBALL

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
D	FACTOR DIFFERENCE
F\$()	FACTOR NAME ARRAY
F()	FACTOR ARRAY
FO	HOME FIELD ADVANTAGE
H\$	NAME OF HOME TEAM
HO()	HEALTH ARRAY
I	INDEX AND ARRAY POINTER
IO()	INTENSITY ARRAY
I1	POINTER TO HOME TEAM
J	INDEX AND ARRAY POINTER
J1	POINTER TO VISITING TEAM
M	NUMBER OF FACTORS
MO()	MORALE ARRAY
N	NUMBER OF TEAMS
T	TOTAL FACTOR POINTS
T\$()	ARRAY OF TEAM NAMES
V\$	NAME OF VISITING TEAM
X\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
CHR\$
DIM
GOSUB
RETURN
TAB

SCORE-BD

Description

This program records the schedules of sports leagues and then analyzes game results. It can be used either for family teams and local sports leagues like Little League and soccer or for professional or college-level athletic league results.

Instructions for Use

Enter the team names and schedules as DATA statements to the program. As games are played, the DATA statements should be modified to reflect their outcomes. The program can then be RUN at any time to review the entries, standings, or remaining games in the schedule. These reports are selected by your choice of options during the program's execution.

Data Entry

All data are entered using DATA statements. These statements should be added to the program in the "DATA ENTRY FOLLOWS" area to replace the sample data. The form of the data is as follows:

The first entry should contain the names of the teams in the league. These entries should be terminated by the word END.

The remaining entries provide information for each scheduled game of the season. The form of this entry differs depending upon whether a specific game has been played or not. Games that are scheduled but not played contain the following:

Home team, visiting team, and date scheduled

The date scheduled must not contain a comma but must begin with an asterisk. After a game has been played, the entry should be changed to remove the asterisk and to add the score of the home team followed by the score of the visiting team. See the sample output. The schedule data are then ended by a DATA statement that contains the word END.

Notes and Things to Check

If a data error occurs, it may be the result of a missing comma or asterisk. Unless an asterisk occurs as the first item of a date, the program will look for scores associated with the game. Therefore all scheduled games should contain an asterisk until their scores are recorded.

Modification Procedures

All input of the schedule is read into program arrays. These arrays can then be used to perform other program functions. Because each option is performed in a separate subroutine, these individual subroutines can be modified without affecting the other program functions. Other subroutines can be added as option choices following line 140, but each subroutine's line number must be added to the "ON O GOTO" statement at line #440, which treats each option separately, even though the GOSUB statement was not used to access the program's subfunctions (subroutines).

```
10 REM          SAVED AT SCORE-BD
20 REM  THIS PROGRAM RECORDS SCHEDULES
30 REM  AND SCORES OF GAMES, IT THEN
40 REM  PRODUCES ANALYSES OF RESULTS
50 REM  *****
60 M=100
70 N=12
80 DIM T1$(M),T2$(M),S1(M),S2(M),D$(M),N$(N),W(N),L(N),T(N),P(N)
90 GOSUB 250      :REM GET INPUT
100 PRINT "THE FOLLOWING OPTIONS ARE AVAILABLE"
110 PRINT TAB(5);"1..PRINT ALL ENTRIES"
120 PRINT TAB(5);"2..PRINT STANDINGS"
130 PRINT TAB(5);"3..PRINT REMAINING SCHEDULE"
140 PRINT TAB(5);"4..PRINT RESULTS"
150 PRINT
160 INPUT "ENTER OPTION DESIRED";O
170 GOSUB 410      :REM PRINT RESULTS
180 REM *****
190 REM  PROGRAM TERMINATION POINT
200 REM *****
210 PRINT
220 PRINT "PROCESSING COMPLETE"
230 STOP
240 REM *****
250 REM          GET INPUT
260 REM *****
270 FOR I=1 TO N
280 READ N$(I)
290 IF N$(I)="END" THEN 310
300 NEXT I
310 N=N-1
320 FOR I=1 TO M
330 READ T1$(I)
340 IF T1$(I)="END" THEN 390
350 READ T2$(I),D$(I)
360 IF LEFT$(D$(I),1)="*" THEN 380
370 READ S1(I),S2(I)
380 NEXT I
390 M=M-1
400 RETURN
410 REM *****
```



```

420 REM                                PRINT RESULTS
430 REM *****
440 ON O GOTO 460,590,900,1020
450 RETURN
460 REM ***** PRINT ALL RECORDS *****
470 PRINT
480 PRINT "HOME";TAB(10);"VISITING";TAB(30);"SCORE"
490 PRINT "TEAM";TAB(12);"TEAM";TAB(20);"DATE";TAB(28);"HT";TAB(34);"V
T"
500 PRINT "-----";TAB(10);"-----";TAB(20);"-----";
510 PRINT TAB(28);"---";TAB(34);"---"
520 FOR I=1 TO M1
530 PRINT T1$(I);TAB(10);T2$(I);TAB(20);D$(I);
540 IF LEFT$(D$(I),1)<>"*" THEN PRINT TAB(28);S1(I);TAB(34);S2(I);
550 PRINT
560 NEXT I
570 PRINT
580 GOTO 1250
590 REM ***** PRINT STANDINGS *****
600 PRINT
610 PRINT TAB(10);"STANDINGS"
620 PRINT
630 PRINT "TEAM";TAB(10);"WON";TAB(15);"LOST";TAB(20);"TIE";TAB(25);"P
CT"
640 PRINT "-----";TAB(10);"---";TAB(15);"---";TAB(20);"---";TAB(25)
;"-----"
650 FOR I=1 TO M1
660 IF LEFT$(D$(I),1)="*" THEN 720
670 GOSUB 1270          'FIND TEAM
680 IF S1(I)>S2(I) THEN W(P1)=W(P1)+1:L(P2)=L(P2)+1
690 IF S1(I)<S2(I) THEN W(P2)=W(P2)+1:L(P1)=L(P1)+1
700 IF S1(I)=S2(I) THEN T(P1)=T(P1)+1:T(P2)=T(P2)+1
710 NEXT I
720 FOR I=1 TO N
730 S(I)=I
740 P(I)=W(I)/(W(I)+L(I)+T(I))
750 NEXT I
760 FOR I=1 TO N-1
770 FOR J=I+1 TO N
780 IF P(I)>P(J) THEN 830
790 IF P(I)=P(J) AND T(I)>=T(J) THEN 830
800 S0=S(I)
810 S(I)=S(J)
820 S(J)=S0
830 NEXT J
840 NEXT I
850 FOR I=1 TO N
860 PRINT N$(S(I));TAB(10);W(S(I));TAB(15);L(S(I));TAB(20);T(S(I));
870 PRINT TAB(25);P(S(I))
880 NEXT I
890 GOTO 1250
900 REM ***** PRINT SCHEDULE *****
910 PRINT
920 PRINT TAB(5);"REMAINING SCHEDULE"
930 PRINT
940 PRINT "HOME";TAB(10);"VISITING";TAB(20);"DATE "
950 PRINT "TEAM";TAB(12);"TEAM";TAB(20);"SCHEDULED"

```

```

960 PRINT "-----";TAB(10);"-----";TAB(20);"-----"
970 FOR I=1 TO M1
980 IF LEFT$(D$(I),1)<>"*" THEN 1000
990 PRINT T1$(I);TAB(10);T2$(I);TAB(20);D$(I)
1000 NEXT I
1010 GOTO 1250
1020 REM ***** TEAM ANALYSIS *****
1030 INPUT "ENTER TEAM NAME";NO$
1040 PRINT
1050 PRINT TAB(5);"TEAM RESULTS"
1060 PRINT
1070 PRINT TAB(20);"OPPONENT"
1080 PRINT TAB(18);"-----"
1090 FOR I=1 TO M1
1100 IF T1$(I)=NO$ OR T2$(I)=NO$ THEN 1120
1110 GOTO 1230
1120 IF LEFT$(D$(I),1)="*" THEN 1220
1130 IF T1$(I)=NO$ THEN 1170
1140 PRINT T2$(I);".....";S2(I);TAB(18);T1$(I);".(H).";S1(I);
1150 IF S2(I)>S1(I) THEN PRINT "WON";
1160 GOTO 1190
1170 PRINT T1$(I);".(H).";S1(I);TAB(18);T2$(I);".....";S2(I);
1180 IF S1(I)>S2(I) THEN PRINT "WON";
1190 IF S1(I)=S2(I) THEN PRINT "TIE";
1200 PRINT
1210 GOTO 1230
1220 PRINT T1$(I);" VS ";T2$(I);" SCHEDULED ";D$(I)
1230 NEXT I
1240 RETURN
1250 RETURN
1260 REM *****
1270 REM          FIND TEAM
1280 REM *****
1290 FOR K=1 TO N
1300 IF T1$(I)=N$(K) THEN P1=K
1310 IF T2$(I)=N$(K) THEN P2=K
1320 NEXT K
1330 RETURN
1340 REM *****
1350 REM          DATA ENTRY FOLLOWS
1360 REM *****
1370 DATA TEAM 1,TEAM 2,TEAM 3,TEAM 4
1380 DATA END
1390 DATA TEAM 1,TEAM 2,OCT 8,5,3
1400 DATA TEAM 3,TEAM 4,OCT 8,2,3
1410 DATA TEAM 4,TEAM 2,OCT 15,2,2
1420 DATA TEAM 3,TEAM 1,OCT 15,2,1
1430 DATA TEAM 1,TEAM 4,OCT 22,3,3
1440 DATA TEAM 3,TEAM 2,OCT 22,1,1
1450 DATA TEAM 2,TEAM 4,OCT 29,3,7
1460 DATA TEAM 1,TEAM 3,OCT 29,1,0
1470 DATA TEAM 2,TEAM 3,*NOV 5
1480 DATA TEAM 4,TEAM 1,*NOV 5
1490 DATA TEAM 4,TEAM 3,*NOV 12
1500 DATA TEAM 2,TEAM 1,*NOV 12
1510 DATA END

```

RUN

THE FOLLOWING OPTIONS ARE AVAILABLE

- 1..PRINT ALL ENTRIES
- 2..PRINT STANDINGS
- 3..PRINT REMAINING SCHEDULE
- 4..PRINT RESULTS

ENTER OPTION DESIRED? 1

HOME TEAM	VISITING TEAM	DATE	SCORE	
			HT	VT
TEAM 1	TEAM 2	OCT 8	5	3
TEAM 3	TEAM 4	OCT 8	2	3
TEAM 4	TEAM 2	OCT 15	2	2
TEAM 3	TEAM 1	OCT 15	2	1
TEAM 1	TEAM 4	OCT 22	3	3
TEAM 3	TEAM 2	OCT 22	1	1
TEAM 2	TEAM 4	OCT 29	3	7
TEAM 1	TEAM 3	OCT 29	1	0
TEAM 2	TEAM 3	*NOV 5		
TEAM 4	TEAM 1	*NOV 5		
TEAM 4	TEAM 3	*NOV 12		
TEAM 2	TEAM 1	*NOV 12		

PROCESSING COMPLETE
Break in 230

RUN

THE FOLLOWING OPTIONS ARE AVAILABLE

- 1..PRINT ALL ENTRIES
- 2..PRINT STANDINGS
- 3..PRINT REMAINING SCHEDULE
- 4.:PRINT RESULTS

ENTER OPTION DESIRED? 2

STANDINGS

TEAM	WON	LOST	TIE	PCT
TEAM 4	2	0	2	.5
TEAM 1	2	1	1	.5
TEAM 3	1	2	1	.25
TEAM 2	0	2	2	0

PROCESSING COMPLETE
Break in 230

RUN
THE FOLLOWING OPTIONS ARE AVAILABLE
1..PRINT ALL ENTRIES
2..PRINT STANDINGS
3..PRINT REMAINING SCHEDULE
4..PRINT RESULTS

ENTER OPTION DESIRED? 3

REMAINING SCHEDULE

HOME TEAM	VISITING TEAM	DATE SCHEDULED
TEAM 2	TEAM 3	*NOV 5
TEAM 4	TEAM 1	*NOV 5
TEAM 4	TEAM 3	*NOV 12
TEAM 2	TEAM 1	*NOV 12

PROCESSING COMPLETE
Break in 230

RUN

THE FOLLOWING OPTIONS ARE AVAILABLE
1..PRINT ALL ENTRIES
2..PRINT STANDINGS
3..PRINT REMAINING SCHEDULE
4..PRINT RESULTS

ENTER OPTION DESIRED? 4

ENTER TEAM NAME? TEAM 2

TEAM RESULTS

	OPPONENT
TEAM 2..... 3	TEAM 1.(H). 5
TEAM 2..... 2	TEAM 4.(H). 2 TIE
TEAM 2..... 1	TEAM 3.(H). 1 TIE
TEAM 2.(H). 3	TEAM 4..... 7
TEAM 2 VS TEAM 3	SCHEDULED *NOV 5
TEAM 2 VS TEAM 1	SCHEDULED *NOV 12

PROCESSING COMPLETE
Break in 230

MAJOR SYMBOL TABLE — SCORE-BD

NAME	DESCRIPTION
D\$()	GAME DATE ARRAY
I	INDEX AND ARRAY POINTER
J	INDEX AND ARRAY POINTER
K	INDEX AND ARRAY POINTER
L()	LOSS RECORD ARRAY
M	MAXIMUM NUMBER OF ENTRIES
M1	ACTUAL NUMBER OF ENTRIES
N	MAXIMUM NUMBER OF TEAMS
N\$()	TEAM NAME ARRAY
NO\$	TEAM TO PRINT
P()	PERCENT ARRAY
P1	POINTER TO HOME TEAM
P2	POINTER TO VISITING TEAM
S()	SORT ARRAY
SO	TEMP SORT VARIABLE
S1()	HOME TEAM SCORE ARRAY
S2()	VISITING SCORE ARRAY
T()	TIE ARRAY
T1\$()	HOME TEAM ARRAY
T2\$()	VISITING TEAM ARRAY
W()	WIN RECORD ARRAY

FUNCTIONS USED

NAME
DIM
GOSUB
LEFT\$
RETURN
TAB

DUESLIST

Description

This program produces a list of money owed compared with money paid. It can be used to record many different items, such as club dues, paper route collections, or charitable pledges.

Instructions for Use

Enter the required information as DATA statements and then RUN the program.

Data Entry

A DATA statement is required for each individual. The format of these statements is as follows:

NAME, AMOUNT OWED, AMOUNT PAID

Notes and Things to Check

Remember to include no commas in the name and to enter the STOP statement as the last item.

Modification Procedures

Data input and totals are computed in the loop at lines 80-150 of the program. You can change the required data formats or computations by modifying this procedure.

Line number 130 is activated whenever money is owed. This causes an asterisk to be printed preceding the name and a counter to be incremented. You can take additional actions by adding statements to this line, preceding each with a colon.

```
10 REM          SAVED AT DUESLIST
20 REM  THIS PROGRAM PRODUCES A LIST OF
30 REM  MONEY OWED VS MONEY PAID
40 REM  *****
50 PRINT CHR$(147)
60 PRINT "NAME";TAB(21);"OWED";TAB(31)"PAID"
70 PRINT "-----"
80 READ N$
90 IF N$="STOP" THEN 170
100 READ O,P
110 T1=T1+P
120 T2=T2+O
130 IF P<O THEN N=N+1:PRINT "**";
```

```

140 PRINT N$;TAB(20);O;TAB(30);P
150 GOTO 80
160 REM *****
170 REM      PROGRAM TERMINATION POINT
180 REM *****
190 T3=T2-T1
200 PRINT "-----"
210 PRINT "TOTAL OWED";TAB(20);T2
220 PRINT "TOTAL PAID";TAB(30);T1
230 PRINT "NUMBER TO PAY";N;"  AMT $";T3
240 PRINT "-----"
250 PRINT
260 STOP
270 REM *****
280 REM      DATA ENTRY FOLLOWS
290 REM *****
300 DATA KIM,14.95,0
310 DATA JIM,12.99,12.99
320 DATA GEORGE,4.95,2.95
330 DATA JUNIOR,24.99,20
340 DATA STOP

```

RUN

NAME	OWED	PAID
*KIM	14.95	0
JIM	12.99	12.99
*GEORGE	4.95	2.95
*JUNIOR	24.99	20

TOTAL OWED	57.88	
TOTAL PAID		35.94
NUMBER TO PAY 3	AMT \$ 21.94	

Break in 260

MAJOR SYMBOL TABLE — DUESLIST

NAME	DESCRIPTION
N	NUMBER OWING
N\$	NAME
O	AMOUNT OWED
P	AMOUNT PAID
T1	TOTAL PAID
T2	TOTAL OWED
T3	TOTAL LEFT TO PAY

FUNCTIONS USED

NAME
CHR\$
TAB

9 For the Commodore 64 User

BUILD-TF

Description

This program can be used to build sequential tape (or disk) files on your computer. It allows you to enter the data, check your entries, and then correct them before they are printed to your storage device. It can be used in a variety of applications for your Commodore 64.

Instructions for Use

RUN the program and then respond to its requests for information. You will be asked to specify the number of items in each record and the name of the file where the information is to be stored before you enter the actual data for the file.

Data Entry

All data are entered at the program's request.

Notes and Things to Check

Be sure that your tape is positioned properly before you run the program. If you are using a disk, you must change the OPEN command at line 110.

Modification Procedures

All data entry, printing, and correction of data are accomplished before any information is placed in the file. The actual writes to the file do not occur until line #230 of the program, when the GOSUB to that subroutine is performed. You can modify the print formats or correction procedures by changing the instructions in appropriate subroutines.

Because data are all entered from the keyboard into an array, you have ample opportunity to modify them before further processing.

```
10 REM          SAVED AT BUILD-TF
20 REM  GENERAL PURPOSE FILE BUILD
30 REM  FOR USE IN BUILDING SEQUENTIAL
40 REM  TAPE AND DISK DATA FILES
50 X$="*****"
60 PRINT "ENTER THE # OF ITEMS PER RECORD (MAX 6) "
70 INPUT N
80 M=100
90 PRINT "ENTER THE NAME OF THE OUTPUT FILE "
100 INPUT F1$
110 OPEN 1,1,1,F1$
120 DIM D$(M,N)
130 PRINT "ENTER THE DATA - 1 RECORD AT A TIME (END WITH A RETURN) "
140 FOR I=1 TO M
150 D$(I,1)=" "
160 GOSUB 770 :REM GET INPUT
170 IF D$(I,1)=" " THEN 190
180 NEXT I
190 I1=I-1
200 PRINT "SHALL I PRINT THE DATA (Y OR N) "
210 INPUT A$
220 IF A$="Y" THEN GOSUB 330:REM PRINT
230 GOSUB 510 :REM PRINT TO FILE
240 REM *****
250 REM  PROGRAM TERMINATION POINT
260 REM *****
270 PRINT
280 PRINT
290 PRINT "PROCESSING COMPLETE"
300 PRINT
310 STOP
320 REM *****
330 REM          PRINT DATA
340 REM *****
350 PRINT
360 PRINT X$
370 PRINT " #";TAB(5);"DATA"
380 PRINT
390 FOR I=1 TO I1
400 PRINT I;TAB(5);
410 FOR J=1 TO N
420 PRINT ">";D$(I,J);"< ";
430 NEXT J
440 PRINT
450 NEXT I
460 PRINT "WOULD YOU LIKE TO CORRECT THE DATA (Y OR N) "
470 INPUT A1$
480 IF A1$="Y" THEN GOSUB 640 :REM CORRECT DATA
490 RETURN
500 REM *****
510 REM          PRINT TO FILE
```

```

520 REM *****
530 FOR I= 1 TO I1
540 PRINT#1,D$(I,1);
550 IF N=1 THEN 590
560 FOR J=2 TO N
570 PRINT#1,CHR$(13);D$(I,J);
580 NEXT J
590 PRINT#1,
600 NEXT I
610 CLOSE 1
620 RETURN
630 REM *****
640 REM      CORRECTION ROUTINE
650 REM *****
660 PRINT "ENTER THE ITEM NUMBER TO CHANGE";
670 INPUT C
680 I=C
690 PRINT "ENTER THE CORRECT DATA"
700 GOSUB 770 ;REM GET INPUT
710 PRINT "ANY MORE CHANGES (Y OR N)";
720 A1$=""
730 INPUT A1$
740 IF A1$="Y" THEN 660
750 RETURN
760 REM *****
770 REM      GET INPUT
780 REM *****
790 ON N GOTO 800,820,840,860,880,900
800 INPUT D$(I,1)
810 GOTO 920
820 INPUT D$(I,1),D$(I,2)
830 GOTO 920
840 INPUT D$(I,1),D$(I,2),D$(I,3)
850 GOTO 920
860 INPUT D$(I,1),D$(I,2),D$(I,3),D$(I,4)
870 GOTO 920
880 INPUT D$(I,1),D$(I,2),D$(I,3),D$(I,4),D$(I,5)
890 GOTO 920
900 INPUT D$(I,1),D$(I,2),D$(I,3),D$(I,4),D$(I,5),D$(I,6)
910 GOTO 920
920 RETURN

```

```

RUN
ENTER THE # OF ITEMS PER RECORD (MAX 6)
? 4
ENTER THE NAME OF THE OUTPUT FILE
? DATA-1
PRESS RECORD & PLAY ON TAPE
OK
ENTER THE DATA - 1 RECORD AT A TIME (END
WITH A RETURN)
? ITEM 1,ITEM 2,ITEM 3,ITEM 4
? ITEM A,ITEM B,ITEM C,ITEM D

```

? ITEM W,ITEM X,ITEM Y,ITEM Z
 ?
 SHALL I PRINT THE DATA (Y OR N)
 ? Y

```
*****
# DATA

1 >ITEM 1< >ITEM 2< >ITEM 3< >ITEM 4<
2 >ITEM A< >ITEM B< >ITEM C< >ITEM D<
3 >ITEM W< >ITEM X< >ITEM Y< >ITEM Z<

WOULD YOU LIKE TO CORRECT THE DATA (Y OR N)
? Y
ENTER THE ITEM NUMBER TO CHANGE? 2
ENTER THE CORRECT DATA
? NEW A,NEW B,NEW C,NEW D
ANY MORE CHANGES (Y OR N)? N
```

PROCESSING COMPLETE

Break in 310

MAJOR SYMBOL TABLE — BUILD-TF

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
A1\$	ANSWER VARIABLE
C	ITEM NUMBER TO CHANGE
D\$()	DATA ARRAY
F1\$	INPUT FILE NAME
I	INDEX AND ARRAY POINTER
II	ACTUAL NUMBER OF ENTRIES
J	INDEX AND ARRAY POINTER
M	MAXIMUM NUMBER OF ENTRIES
N	NUMBER OF ITEMS/RECORD
X\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
CHR\$
CLOSE
DIM
GOSUB
OPEN
RETURN

PRINT-TF

Description

This program reads any sequential data file from tape (or disk) and prints the results to the screen. It can be used for a variety of purposes to assist you in the operation of your Commodore 64 and the development of programs by providing a quick method of looking at your program's effect upon a data file.

Instructions for Use

RUN the program and respond to its requests for information. Because the program requires that each record contain the same amount of information, you may find it desirable to specify a one for this answer if the records on your file have different numbers of items in each.

Data Entry

All data are entered from the data file or in response to the program's request.

Notes and Things to Check

If you have difficulty getting the appropriate data from the file, you may find it necessary to specify the file as only having one item per record. If you are using the program to retrieve disk files, you will need to change the OPEN statement at line #120 of the program.

Modification Procedures

Not applicable to this program.

```
10 REM      SAVED AT PRINT-TF
20 REM      GENERAL PURPOSE SEQUENTIAL FILE
30 REM      PRINT UTILITY FOR USE IN
40 REM      PRINTING SEQUENTIAL TAPE AND
50 REM      DISK DATA FILES
60 X$="*****"
70 PRINT "ENTER THE # OF ITEMS PER RECORD (MAX 6) "
80 INPUT N
90 M=100
100 PRINT "ENTER THE NAME OF THE INPUT FILE";
110 INPUT F1$
120 OPEN 1,1,0,F1$
130 DIM D$(M,N)
140 GOSUB 450      :REM GET INPUT
```

```

150 I1=I-1
160 PRINT "SHALL I PRINT THE DATA (Y OR N) "
170 INPUT A$
180 IF A$="Y" THEN GOSUB 290:REM PRINT
190 IF A$<>"Y" THEN PRINT I1;"RECORDS WERE FOUND"
200 REM *****
210 REM   PROGRAM TERMINATION POINT
220 REM *****
230 PRINT
240 PRINT
250 PRINT "PROCESSING COMPLETE"
260 PRINT
270 STOP
280 REM *****
290 REM           PRINT DATA
300 REM *****
310 PRINT
320 PRINT X$
330 PRINT " #";TAB(5);"DATA"
340 PRINT
350 FOR I=1 TO I1
360 PRINT I;TAB(5);
370 FOR J=1 TO N
380 PRINT ">";D$(I,J);"< ";
390 NEXT J
400 PRINT
410 NEXT I
420 PRINT "EOF"
430 RETURN
440 REM *****
450 REM           GET INPUT
460 REM *****
470 FOR I=1 TO M
480 FOR J=1 TO N
490 IF ST<>0 THEN 530
500 INPUT#1,D$(I,J)
510 NEXT J
520 NEXT I
530 CLOSE 1
540 RETURN

```

```

RUN
ENTER THE # OF ITEMS PER RECORD (MAX 6)
? 4
ENTER THE NAME OF THE INPUT FILE? DATA-1

PRESS PLAY ON TAPE
OK
SHALL I PRINT THE DATA (Y OR N)
? Y

```

DATA

1 >ITEM 1< >ITEM 2< >ITEM 3< >ITEM 4<

2 >NEW A< >NEW B< >NEW C< >NEW D<

3 >ITEM W< >ITEM X< >ITEM Y< >ITEM Z<

EOF

PROCESSING COMPLETE

Break in 270

MAJOR SYMBOL TABLE — PRINT-TF

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
D\$()	DATA ARRAY
F1\$	NAME OF INPUT FILE
I	INDEX AND ARRAY POINTER
I1	NUMBER DATA ITEMS READ
J	INDEX AND ARRAY POINTER
M	MAXIMUM NUMBER ALLOWED
N	NUMBER ITEMS PER RECORD
X\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
CLOSE
DIM
GOSUB
OPEN
RETURN
ST

TAPE-LIB

Description

This program provides a means of recording and retrieving the names and storage locations of your program and data files. It can assist you to remember this information. It can also be slightly modified to perform the same function for family members' music tapes.

Instructions for Use

Tapes and the files stored on them should be entered as DATA statements as soon as it is convenient. The program can then be RUN at any time to produce either a list of all tapes and files or information about individual tapes or files.

Data Entry

All data are entered as DATA statements. These data have two different formats, and the program looks for two types of records in the data. The first type is the tape name. When the program finds any entry that is not an F, it treats that record as the master heading for a tape. The follow-records (those beginning with an F) are treated as file entries contained on that tape. This continues until the next non-F record occurs. The format of the file record is:

DATA F, file name, file type, tape location

The format for the tape master record is:

DATA tape name

The last data entry must be END, to terminate the program's operation.

Notes and Things to Check

Be sure that your choice of tape names does not include a comma and that each file entry begins with an F.

Modification Procedures

Note that the print of all items is accomplished during data input unless a search has been requested. If a search for either a specific tape or a specific file has been requested, the program does not print the item until it checks to see if it matches the selection criteria of the "SEARCH ROUTINE," at lines 380-600 of the program.

```

10 REM          SAVED AT TAPE-LIB
20 REM GENERAL PURPOSE PROGRAM FOR
30 REM MAINTAINING PROG AND DATA TAPES
40 REM FOR YOUR COMMODORE 64
50 REM *****
60 C0=1
70 M=1000
80 PRINT "SHALL I PRINT ALL ITEMS (Y OR N)";
90 INPUT A0$
100 IF A0$="Y" THEN 180
110 PRINT "SHALL I SEARCH FOR A TAPE(T) OR FILE(F) "
120 INPUT A1$
130 PRINT "ENTER ITEM TO SEARCH FOR";
140 INPUT X$
150 PRINT
160 PRINT
170 C0=0
180 REM *****PROCESSING AREA*****
190 FOR I=1 TO M
200 READ T$
210 IF T$="END" THEN 310
220 IF T$<>"F" THEN GOSUB 620 :REM PRINT MASTER ITEMS
230 IF T$<>"F" THEN 290
240 READ A$,P$,L$
250 IF A0$<>"Y" THEN 280
260 PRINT TAB(5);P$;TAB(18);A$;TAB(30);L$
270 GOTO 290
280 GOSUB 380 :REM SEARCH ROUTINE
290 NEXT I
300 REM *****
310 REM PROGRAM TERMINATION POINT
320 REM *****
330 IF C0<>0 THEN 350
340 PRINT "ITEM NOT FOUND"
350 PRINT
360 STOP
370 REM *****
380 REM SEARCH ROUTINE
390 REM *****
400 IF A1$<>"T" THEN 530
410 IF T$<>"F" THEN 600
420 IF T1$<>X$ THEN 600
430 IF C>0 THEN 490
440 PRINT "TAPE - ";T1$
450 PRINT
460 IF C0>0 THEN 490
470 PRINT TAB(5);"FILE TYPE";TAB(15);"NAME";TAB(30);"LOCATION"
480 PRINT TAB(5);"-----";TAB(15);"-----";TAB(30);"-----"
490 PRINT TAB(5);A$;TAB(15);P$;TAB(30);L$
500 C0=C0+1
510 C=C+1
520 GOTO 600
530 IF A1$<>"F" THEN 600
540 IF A$<>X$ THEN 600
550 IF C0>0 THEN 590
560 PRINT "FILENAME";TAB(12);"FILE TYPE";TAB(24);"TAPE";TAB(32);"LOC"
570 PRINT "-----";TAB(12);"-----";TAB(24);"-----";TAB(32);"-----"

```



```

580 C0=C0+1
590 PRINT A$;TAB(12);P$;TAB(24);T1$;TAB(32);L$
600 RETURN
610 REM *****
620 REM PRINT MASTER ITEM
630 REM *****
640 T1$=T$
650 C=0
660 IF A0$<>"Y" THEN 720
670 PRINT
680 PRINT "TAPE - ";T$
690 PRINT
700 PRINT TAB(5);"FILE TYPE";TAB(18);"NAME";TAB(30);"LOCATION"
710 PRINT TAB(5);"-----";TAB(18);"----";TAB(30);"-----"
720 RETURN
730 REM *****
740 REM PLACE DATA STATEMENTS HERE
750 REM *****
760 DATA 100
770 DATA F,X,DATA,5
780 DATA F,Y,PROGRAM,50
790 DATA 101
800 DATA F,DF,DATA,1
810 DATA F,PF,PROGRAM,20
820 DATA END

```

```

RUN
SHALL I PRINT ALL ITEMS (Y OR N)? N
SHALL I SEARCH FOR A TAPE(T) OR FILE(F)
? T
ENTER ITEM TO SEARCH FOR? 100

```

TAPE - 100

FILE TYPE	NAME	LOCATION
-----	----	-----
X	DATA	5
Y	PROGRAM	50

Break in 360

```

RUN
SHALL I PRINT ALL ITEMS (Y OR N)? N
SHALL I SEARCH FOR A TAPE(T) OR FILE(F)
? F
ENTER ITEM TO SEARCH FOR? PF

```

FILENAME	FILE TYPE	TAPE	LOC
PF	PROGRAM	101	20

Break in 360

RUN
SHALL I PRINT ALL ITEMS (Y OR N)? Y

TAPE - 100

FILE TYPE	NAME	LOCATION
DATA	X	5
PROGRAM	Y	50

TAPE - 101

FILE TYPE	NAME	LOCATION
DATA	DF	1
PROGRAM	PF	20

Break in 360

MAJOR SYMBOL TABLE — TAPE-LIB

NAME	DESCRIPTION
A\$	FILE TYPE FROM DATA
AO\$	ANSWER VARIABLE
A1\$	ITEM TO SEARCH
C	COUNT VARIABLE
CO	COUNT VARIABLE
I	INDEX AND ARRAY POINTER
L\$	LOCATION FROM DATA
M	MAXIMUM ENTRIES ALLOWED
P\$	PROGRAM/FILE NAME
T\$	TAPE NAME
T1\$	TAPE NAME SAVED
X\$	ITEM TO SEARCH FOR

FUNCTIONS USED

NAME
GOSUB
RETURN
TAB

HEX-DEC

Description

Since many of the Commodore manuals reference memory locations using the hexadecimal number system, this program can be used to convert hexadecimals to the decimal numbers we normally use (which are also required for some program statements).

Instructions for Use

RUN the program and enter the hexadecimal numbers that you want to convert. A blank entry will terminate the program.

Data Entry

Data are entered at the program's request.

Notes and Things to Check

Hexadecimal numbers can use only the characters 0,1,2,3,4,5,6,7,8,9,A,B,C,D,E, or F. These are the only valid entries for conversion.

Modification Procedures

Note that any hexadecimal number is read as a string and converted by the IF statements at lines 140-200 of the program.

```
10 REM      SAVED AT HEX-DEC
20 REM      THIS PROGRAM ACCEPTS HEX NUMBER
30 REM      INPUTS AND PRINTS DECIMAL EQUIVS
40 REM      *****
50 DIM X$(5)
60 H$=""
70 INPUT "HEXADECIMAL NUMBER";H$
80 IF H$="" THEN 240
90 X=0
100 L=LEN(H$)
110 FOR I=1 TO L
120 J=L-I+1
130 X$(I)=MID$(H$,J,1)
140 IF X$(I)="A" THEN X$(I)="10"
150 IF X$(I)="B" THEN X$(I)="11"
160 IF X$(I)="C" THEN X$(I)="12"
170 IF X$(I)="D" THEN X$(I)="13"
180 IF X$(I)="E" THEN X$(I)="14"
190 IF X$(I)="F" THEN X$(I)="15"
```

```

200 X=X+VAL(X$(I))*16^(I-1)
210 NEXT I
220 PRINT "THE DECIMAL VALUE IS ";X
230 GOTO 60
240 REM *****
250 PRINT
260 PRINT "PROCESSING COMPLETE"
270 PRINT
280 STOP

```

```

RUN
HEXADECIMAL NUMBER? D01A
THE DECIMAL VALUE IS 53274
HEXADECIMAL NUMBER? F
THE DECIMAL VALUE IS 15
HEXADECIMAL NUMBER? FF
THE DECIMAL VALUE IS 255
HEXADECIMAL NUMBER? 0288
THE DECIMAL VALUE IS 648
HEXADECIMAL NUMBER?

```

PROCESSING COMPLETE

Break in 280

MAJOR SYMBOL TABLE — HEX-DEC

NAME	DESCRIPTION
H\$	HEX NUMBER INPUT
I	INDEX POINTER
J	INDEX POINTER
L	LENGTH OF H\$
X	DECIMAL VALUE OF HEX NUMBER
X\$()	ARRAY OF HEX NUMBERS INPUT

FUNCTIONS USED

NAME
DIM
LEN
MID\$
VAL

COLUMN

Description

This program provides you with a general-purpose capability of printing formatted reports from data input either from data files or from the keyboard. You can use it in many different applications to produce nicely formatted reports without writing a new program each time.

Instructions for Use

RUN the program and respond to its requests. The data entered as "HEADING/PROMPTS" are used both as headings for the report and as prompts for your data input if you enter your data directly to the program. If you use file entry, this program will not change the data file.

Data Entry

The data to be printed can be entered directly from the keyboard or read from an existing data file.

Notes and Things to Check

The Commodore 64 screen display is only 40 characters wide. To maintain correct formatting, then, your heading choices and print locations should not print beyond 40 characters.

Modification Procedures

The program is separated into three subroutines: "GET DATA" retrieves information from a file; "ACCEPT DATA" gets the data from the keyboard; and "PRINT DATA" provides the output. For example, if you want to provide your formatted output to a printer, you should OPEN the printer file in the "PRINT DATA" subroutine and then change the appropriate PRINT statements in that routine to send the output to the printer.

```
10 REM      SAVED AT COLUMN
20 REM      GENERAL PURPOSE SEQUENTIAL
30 REM FILE UTILITY TO PRINT DATA FILES
40 REM OR KEYBOARD DATA IN COLUMN
50 REM FORM
60 X$="*****"
70 PRINT "ENTER THE # OF ITEMS PER RECORD (MAX 6)"
80 INPUT N
90 M=100
```

```

100 DIM D$(M,N),T$(N),T(N)
110 PRINT "ENTER THE HEADINGS/PROMPTS FOR EACH"
120 FOR I=1 TO N
130 PRINT "FOR FIELD #";I;
140 INPUT T$(I)
150 NEXT I
160 PRINT
170 PRINT "IS THE INPUT TO COME FROM A FILE(Y OR N)"
180 INPUT A$
190 PRINT
200 IF A$<>"Y" THEN 260
210 PRINT "ENTER THE NAME OF THE INPUT FILE"
220 INPUT F1$
230 OPEN 1,1,0,F1$
240 GOSUB 670 :REM GET INPUT
250 GOTO 270
260 GOSUB 780 :REM ACCEPT DATA
270 I1=I-1
280 PRINT "SHALL I PRINT THE DATA (Y OR N)"
290 INPUT A$
300 IF A$="Y" THEN GOSUB 410:REM PRINT
310 IF A$<>"Y" THEN PRINT I1;"RECORDS WERE FOUND"
320 REM *****
330 REM PROGRAM TERMINATION POINT
340 REM *****
350 PRINT
360 PRINT
370 PRINT "PROCESSING COMPLETE"
380 PRINT
390 STOP
400 REM *****
410 REM PRINT DATA
420 REM *****
430 PRINT
440 PRINT "ENTER THE TAB POSITIONS FOR EACH HEADING"
450 FOR I=1 TO N
460 PRINT "FOR HEADING #";I;
470 INPUT T(I)
480 NEXT I
490 PRINT CHR$(147)
500 PRINT X$
510 FOR I=1 TO M
520 PRINT TAB(T(I));T$(I);
530 NEXT I
540 PRINT
550 PRINT
560 PRINT
570 FOR I=1 TO I1
580 FOR J=1 TO N
590 PRINT TAB(T(J));D$(I,J);
600 NEXT J
610 PRINT
620 NEXT I
630 PRINT X$
640 PRINT "EOF"
650 RETURN
660 REM *****

```

```

670 REM          GET INPUT
680 REM *****
690 FOR I=1 TO M
700 FOR J=1 TO N
710 IF ST<>0 THEN 750
720 INPUT#1,D$(I,J)
730 NEXT J
740 NEXT I
750 CLOSE 1
760 RETURN
770 REM *****
780 REM          ACCEPT DATA
790 REM *****
800 PRINT "ENTER DATA FOR THE PROMPTS"
810 PRINT "A RETURN IN THE FIRST FIELD"
820 PRINT "WILL TERMINATE ENTRY"
830 PRINT
840 PRINT "ENTER:";
850 FOR I=1 TO M
860 FOR J=1 TO N
870 PRINT TAB(7);T$(J);
880 INPUT D$(I,J)
890 IF J=1 AND D$(I,J)=" THEN 930
900 NEXT J
910 PRINT
920 NEXT I
930 RETURN

```

```

RUN
ENTER THE # OF ITEMS PER RECORD (MAX 6)
? 4
ENTER THE HEADINGS/PROMPTS FOR EACH
FOR FIELD # 1 ? HEADING1
FOR FIELD # 2 ? HEADING2
FOR FIELD # 3 ? HEADING3
FOR FIELD # 4 ? HEADING4

IS THE INPUT TO COME FROM A FILE(Y OR N)
? Y

ENTER THE NAME OF THE INPUT FILE
? DATA-1
PRESS PLAY ON TAPE
OK
SHALL I PRINT THE DATA (Y OR N)
? Y

```

ENTER THE TAB POSITIONS FOR EACH HEADING

FOR HEADING # 1 ? 1
FOR HEADING # 2 ? 10
FOR HEADING # 3 ? 20
FOR HEADING # 4 ? 30

HEADING1 HEADING2 HEADING3 HEADING4

ITEM 1	ITEM 2	ITEM 3	ITEM 4
NEW A	NEW B	NEW C	NEW D
ITEM W	ITEM X	ITEM Y	ITEM Z

EOF

PROCESSING COMPLETE

Break in 390

RUN

ENTER THE # OF ITEMS PER RECORD (MAX 6)
? 2
ENTER THE HEADINGS/PROMPTS FOR EACH
FOR FIELD # 1 ? HEADING 1
FOR FIELD # 2 ? HEADING 2

IS THE INPUT TO COME FROM A FILE(Y OR N)

? N

ENTER DATA FOR THE PROMPTS
A RETURN IN THE FIRST FIELD
WILL TERMINATE ENTRY

ENTER:HEADING 1? ITEM 1 OF 1
HEADING 2? ITEM 1 OF 2

HEADING 1? ITEM 2 OF 1
HEADING 2? ITEM 2 OF 2

HEADING 1?

SHALL I PRINT THE DATA (Y OR N)
? Y

ENTER THE TAB POSITIONS FOR EACH HEADING

FOR HEADING # 1 ? 5
FOR HEADING # 2 ? 20


```

*****
HEADING 1      HEADING 2

ITEM 1 OF 1    ITEM 1 OF 2
ITEM 2 OF 1    ITEM 2 OF 2
*****
EOF

```

PROCESSING COMPLETE

Break in 390

MAJOR SYMBOL TABLE — COLUMN

NAME	DESCRIPTION
A\$	ANSWER VARIABLE
D\$()	DATA ARRAY
F1\$	INPUT FILE NAME
I	INDEX AND ARRAY POINTER
11	ACTUAL NUMBER OF ENTRIES
J	INDEX AND ARRAY POINTER
M	MAXIMUM NUMBER OF ENTRIES
N	NUMBER OF ITEMS/RECORDS
T\$()	TITLE HEADINGS/PROMPT ARRAY
T()	TAB POSITION ARRAY
X\$	LINE OF ASTERISKS

FUNCTIONS USED

NAME
CHR\$
CLOSE
DIM
GOSUB
OPEN
RETURN
ST

Appendix A

Language Features Used

The programs in this book were written and tested on a Commodore 64 Computer System equipped with only the standard features available on that machine. Modifying these programs to run on other machines, however, should not be a difficult task. To assist you in this task, some of the less standard functions and statements are explained in this appendix.

CHR\$() Although most machines use this function to convert the ASCII code to its character equivalent, certain uses of this function in these programs should be explained:

CHR\$(147) is used frequently. Printing this value to the screen causes a clear-screen activity similar to the CLS function in some other common Microsoft-based systems.

CHR\$(13) is used to place a carriage return between items stored on tape (or disk) files.

CHR\$(145) is used to move the cursor one line upward.

DIM This statement creates an array for the variable of the size specified within the parentheses. For example, DIM A(5) creates a variable "A" with five positions. Variables not explicitly dimensioned in Commodore BASIC are automatically dimensioned as eleven. Care has been taken, however, explicitly to dimension all variables in the programs.

GET This statement is used to get one character from the file specified. If no file number is used, the GET is applied to the keyboard.

INPUT# This statement retrieves a data item from the file that is OPENed as the file number that follows the #.

LEFT\$(string, integer) This statement retrieves the leftmost integer number of characters from the string variable specified.

LEN This returns the number of characters in the string expression that is provided in its parentheses.

MID\$(string, start, number) This returns the number of characters specified from the string, beginning at the start position in the string.

OPEN The Commodore OPEN statement has many variations. In this book, its use has been confined to OPENing tape files. Two formats are used. Their meanings are as follows:

OPEN 1,1,1,filename

This statement opens the tape cassette as file#1, with write permission (for output). It is equivalent to such commands of other systems as OPEN "O",1,filename, or OPEN filename FOR OUTPUT AS #1.

OPEN 1,1,0,filename

This statement also opens the tape cassette as file#1, but in this case the file is opened for input only.

PRINT# This command, like its INPUT# equivalent, writes the data item to the file number specified. This file number must previously have been OPENed for output.

RND This function returns a random floating-point number from .0 to 1.0.

SPC This function prints the number of spaces specified. For example, SPC(40) prints forty spaces to the output device.

ST This function returns the completion status from the last input/output operation. It is used in these programs to check its value against zero, since a nonzero condition indicates an end-of-file or some other error condition. Its equivalent in other systems is normally IF EOF(file#) THEN . . .

TI This function returns the time in 1/60-second increments since the system was powered up. It is occasionally used to provide a timer delay for program activities.

Appendix B

Converting Programs for Floppy Disk Usage

The programs in this book have been written to use a data cassette for file storage. Many of the programs, however, can effectively use a Commodore 64 disk drive if one is available. The modifications necessary to change the input/output from a data cassette to a floppy disk drive can be easily made through a change in the OPEN command that appears in the program listing.

Formats of the OPEN Statement

To determine how to change the OPEN statement, it is necessary to understand the meaning of the various items included in the statement.

For cassette operations the format of the OPEN statement is

OPEN file#,device,activity,"filename"

where

File# is a number from 1 to 127 used in INPUT# and OUTPUT# statements to refer to the file.

Device indicates the Input/Output device that will be used; a 1 indicates the data cassette.

Activity specifies whether we are reading or writing to the device. A 0 indicates that we are reading from it, a 1 indicates that we are writing to it, and a 2 indicates that we are writing to it and storing an End-of-Tape mark when we close the file.

Filename is a name, of our choice, that specifies the name under which the file is to be stored (if we are writing) or the name to search for if we are reading from the tape. This entry must be enclosed in quotation marks.

The following examples illustrate the OPEN command as it applies to tape files:

OPEN 1,1,0,"FILE1" Opens the file named FILE1 for input (reading) to the program. Inputs from the file use INPUT#1 statements.

OPEN 3,1,1,"FILE2" Opens the file named FILE2 for output (writing) from the program. Outputs to the file use PRINT#3 statements.

For floppy disk operations, the OPEN statement is changed considerably. Its format is as follows:

OPEN file#,device,channel#,"0:filename,type,activity"

where:

File#, as in the cassette format, is a number from 1 to 127.

Device indicates the input/output device that will be used; an 8 indicates the floppy disk.

Channel# is a number from 2 to 15 that identifies further the input/output path of the data. For data file usage, you should not use the number 15. This number is used for command purposes.

Filename is the name that we will use to identify the file, or the name under which it was previously stored.

Type refers to the type of file we are using. All programs in this book use sequential files, type SEQ.

Activity refers to the direction of data flow. An R indicates that the program is to read from the file and a W specifies that the program is to write to the file.

The following examples illustrate the use of the OPEN command for disk:

OPEN 1,8,3,"0:FILE1,SEQ,R" opens the file named FILE1 for input (reading) by the program using INPUT#1 statements.

OPEN 3,8,3,"0:FILE2,SEQ,W" opens the file named FILE2 for output (writing) from the program using PRINT#3 statements.

Converting Programs to Disk Operations

To convert any program in this book to disk files instead of data cassette files, you simply find and change the OPEN statements in that program. The OPEN statement must be changed to reflect the differences in format illustrated above. To clarify this, the following paragraphs indicate the modifications necessary to change the programs BUILD-TF and PRINT-TF from data cassette to disk file usage.

BUILD-TF This program is used to build sequential files. The sample listing can be used as is for data cassette storage, but it must be modified for disk. The OPEN statement appears at line # 110 as OPEN

1,1,1,F1\$. This statement opens the file specified by the variable F1\$, for output as file number 1. To change this program to disk output this line must be changed to

```
110 OPEN 1,8,5,"0:"+F1$+",SEQ,W"
```

PRINT-TF This program is used to read sequential files. In this case, the OPEN statement occurs at line # 120 of the program as OPEN 1,1,0,F1\$. This statement opens the file specified by F1\$ for input by the program. To change this program to disk input, this line must be changed to

```
120 OPEN 1,8,5,"0:"+F1$+",SEQ,R"
```

In these examples, because the file name was provided in a variable name, we found it necessary to concentrate the other required entries to it by using the + operator.

Commodore 64™ Programs for the Home

Charles D. Sternberg

Contains over 40 BASIC programs ranging from game and recreational programs, to educational tutor programs, to programs for home financial management. Program applications for every member of the family.

Each program is documented with a description of its functions and operation, a listing in Commodore 64 BASIC, a symbol table, sample data, and a sample run. And all programs can be easily modified.

Another Book of Interest...

Basic Commodore 64™ BASIC

James S. Coan

Here's the key to using and enjoying the Commodore 64 microcomputer. A fast-reading guide to computer operation and BASIC programming that is suitable for use at home or in the classroom. The approach is simple and direct. Start with short computer programs, master them quickly, add a new command, and watch as the desired effect on the program is created and illustrated. Move on to another capability. Programs are divided into manageable segments — most will fit on a single screen. Special features and advanced programming techniques are clearly explained. For example, the screen editor, immediate mode execution, and utilization of memory for graphics programs are covered. #6456-x, paper, 256 pages.



HAYDEN BOOK COMPANY, INC.
Hasbrouck Heights, New Jersey

ISBN 0-8104-5176-X