

Passo a Passo

MICROSOFT

APRESENTAÇÃO

O Passo a Passo é um curso de BASIC, destinado a principiantes. Tanto na versão em fita como em disquete, o curso se compõe de dez aulas divididas em duas partes diferentes cada uma. As aulas são auto-explicativas e seguidas de um teste de avaliação. Você deve acompanhá-las através do Tutorial, para obter maiores esclarecimentos.

Versão Fita

As aulas e testes de 1 a 5 encontram-se do lado 1 da fita, e as aulas e testes de 6 a 10 encontram-se do lado 2.

Para carregar, no computador o menu, do lado 1 ou do lado 2 da fita, conecte o cabo apropriado nas tomadas EAR do computador e do gravador, em seguida conecte o cabo correspondente no REMOTE CONTROL A do computador e na tomada REM do gravador.

Ligue o computador, digite MOTOR1: LOADT (RETURN) e pressione a tecla PLAY do gravador.

Após a carga do menu no computador, você poderá optar por qualquer das aulas ou testes relacionados no menu. Por meio de controle remoto, o computador procurará automaticamente a opção na fita.

Sempre que houver parada do sistema visual, pressione (RETURN) para dar continuidade.

Terminada uma aula ou teste, não será preciso rebobinar a fita para obter a sequência das aulas. Se você preferir assistir a qualquer outra aula que não a da sequência, desconecte o cabo para controle remoto, rebobine a fita e carregue o menu novamente. Se você desejar interromper o curso em determinada aula, para retorná-la posteriormente no mesmo ponto, não rebobine a fita. O curso prosseguirá normalmente, sendo a fita lida a partir do ponto onde se deu a interrupção através do comando MOTOR1: LOADT (RETURN).

Versão Disquete

Insira o disquete na unidade de disco, para a leitura do lado 1. Ligue o computador.

Após a carga de parte do programa, você obterá uma tela de apresentação, seguida de um menu das aulas e testes do curso. Selecione a opção a ser carregada no computador.

Sempre que houver parada do sistema visual, pressione (RETURN), para dar continuidade.

Ao final de cada aula, você deverá desligar e ligar o computador, para selecionar outra opção do menu.

TABELA DE CONTEÚDO

■ AULA 1

- PARTE 1 - Aspectos gerais do computador e da Linguagem BASIC 1-2
- PARTE 2 - Instrução PRINT nas operações aritméticas 2-4

■ AULA 2

- PARTE 1 - Programas, Comandos RUN e LIST 5-7
- PARTE 2 - Correção de erros, Comandos NEW, END e STOP 7-9

■ AULA 3

- PARTE 1 - Instrução LET 10-12
- PARTE 2 - Instrução INPUT, mensagem REENTRE 12-14

■ AULA 4

- PARTE 1 - Instruções GOTO, CTRL-C, STOP 15-18
- PARTE 2 - IF...THEN, Sumário dos sinais comparativos 18-21

■ AULA 5

- PARTE 1 - Contadores, REM 21-24
- PARTE 2 - Funções Bibliotecárias: INT, ABS, RND, SQS, SEN, LOG, etc 24-28

■ AULA 6

- PARTE 1 - Comandos de controle de tela: HOME, PRINT, VTAB, HTAB 29-31
- PARTE 2 - Variáveis numéricas e STRINGS 31-33

■ AULA 7

- PARTE 1 - GOSUB e RETURN 34-35
- PARTE 2 - READ, DATA e RESTORE 36-38

■ AULA 8

- PARTE 1 - LOOPS FOR...NEXT, STEP 39-40
- PARTE 2 - Vetores, Instrução DIM 41-42

■ AULA 9

- PARTE 1 - LOOPS compostos 43-44
- PARTE 2 - Desenhos: GR Instruções: PLOT, HLIN, VLIN, COLOR, TEXT 44-47

■ AULA 10

- PARTE 1 - Linhas de múltiplas instruções 48-49
- PARTE 2 - Exponenciação e Notação Científica 49-53



AULA 1 PARTE 1

■ PRESSIONE RETURN(CR)

VISUAL 1

Se você pudesse olhar dentro da memória de seu computador, você veria tudo em código. Algo parecido com o que está sendo apresentado na tela. Esta é a única linguagem compreendida pelo computador, por isso esta linguagem é chamada de LINGUAGEM DE MÁQUINA (ASSEMBLER).

■ RETURN(CR)

VISUAL 2

Apresenta um programa em LINGUAGEM DE MÁQUINA. São séries de instruções que o computador segue para realizar certa tarefa. Como é nesta linguagem que o computador compreende, essas instruções não significam nada para a maioria de nós. Para facilitar a tradução das outras linguagens foi criado um INTERPRETADOR.

■ RETURN(CR)

VISUAL 3

O INTERPRETADOR está situado na memória do computador. Aqui é mostrado como ele codifica a linguagem BASIC para linguagem de máquina.

■ RETURN(CR)

■ RETURN(CR)

VISUAL 4

Como é mostrado, a linguagem BASIC tem o significado de BEGINNERS ALL PURPOSE SYMBOLIC INSTRUCTION CODE.

BASIC foi a primeira linguagem de computação para programadores iniciantes, porém hoje é a linguagem mais utilizada em pequenos computadores, isto é - em microcomputadores.

■ RETURN(CR)

VISUAL 5

Mostra algumas instruções e comandos utilizados em BASIC.

Responda às perguntas propostas.

■ RETURN(CR)

VISUAL 6

Este curso vai ensinar a você os comandos e instruções utilizados em BASIC para a efetiva programação de seu Microcomputador.

■ RETURN(CR)

VISUAL 7

Outros comandos e instruções em BASIC.

■ RETURN(CR)

RETURN(CR)

Se você errar na maneira de digitar uma instrução em BASIC, o computador - fornecerá uma mensagem de erro; ? GRAFIA ERRO ou SINTAXE ERRO. Erro de sintaxe significa que as letras foram dispostas de modo errado.

RETURN(CR)

VISUAL 9

O primeiro comando que vamos apresentar é PRINT. PRINT como o nome diz, imprime tipos, letras ou números na tela do computador. Vamos ver como isso é feito:

RETURN(CR)

O computador digita na tela a instrução PRINT

■ RETURN(CR)

Então aparece aspas

RETURN(CR)

Depois aparece digitado o texto que se quer ver na tela

RETURN(CR)

E, finalmente fecha aspas

RETURN(CR)

Como pôde ser observado o texto que foi colocado entre aspas é apresentado na tela.

RETURN(CR)

Responda às perguntas propostas

RETURN(CR)

VISUAL 10

Agora, abra o Caderno de Exercícios que acompanha este curso, leia o resumo e execute os exercícios.

RETURN(CR)

RETURN(CR)

Desligue o equipamento.

Após feito os exercícios, ligue o Microcomputador e digite o número 2 para carregar AULA 1 PARTE 2

[Faint, illegible text from the reverse side of the page]

AULA 1 PARTE 2

Na primeira aula você aprendeu utilizar o comando PRINT para apresentar correntes de letras ou outros caracteres. Nesta aula será usado o comando PRINT para executar operações aritméticas.

RETURN(CR)

VISUAL 1

O computador faz qualquer operação matemática e mostra os resultados se vo

cê utilizar a instrução PRINT

Primeiro a instrução PRINT aparece na tela

■ RETURN(CR)

E a resposta é apresentada.

■ RETURN(CR)

Responda às perguntas

■ RETURN(CR)

O computador, entretanto não faz operações aritméticas, quando estas operações estiverem entre aspas.

■ RETURN(CR)

Qualquer caracter colocado entre aspas é tratado como se fosse letra, e é impresso exatamente como foi dado na entrada.

■ RETURN(CR)

Responda às perguntas propostas.

■ RETURN(CR)

VISUAL 2

■ RETURN(CR)

Apesar do computador utilizar os mesmos símbolos que nós utilizamos para a operação de adição e subtração, um asterisco * é usado para multiplicação e a barra inclinada / para divisão.

■ RETURN(CR)

Responda às perguntas propostas.

■ RETURN(CR)

VISUAL 3

Alguns problemas de matemática, misturam adição, subtração, multiplicação e, divisão. Você precisa saber como o computador executa as operações. Todas as operações de multiplicação e divisão são executadas com prioridade. Neste exemplo a operação de multiplicação é executada:

■ RETURN(CR)

$8 * 2$ é substituído por 16. Em seguida é feita a divisão:

■ RETURN(CR)

$16 / 8$ é substituído por 2. Finalmente adição e subtração são executadas:

■ RETURN(CR)

Oferecendo como resultado final, 5.

■ RETURN(CR)

Responda às perguntas propostas.

VISUAL 4

Quando parêntesis são utilizados em um problema, todas as operações entre parêntesis são executadas com prioridade.

■ RETURN(CR)

■ RETURN(CR)

VISUAL 1

Na aula anterior você aprendeu a fazer uso do comando PRINT. Um comando é - um único passo. Entretanto, a maioria das tarefas executadas pelo computador, requer vários passos.

■ RETURN(CR)

Você agora está visualizando uma amostra de programa. Um programa consiste em uma sequência de passos, cada um dos quais é chamado de LINHA DE INSTRUÇÃO. E, para cada expressão é designada uma linha com um número.

■ RETURN(CR)

Responda às perguntas propostas.

Os comandos que você utilizou na aula anterior são executados no momento - em que você pressionar a tecla RETURN(CR).■

Uma expressão em um programa somente é executada quando um programa inteiro for executado.

VISUAL 2

■ RETURN(CR)

Observe o programa exposto. Para que este programa seja executado, digitamos a palavra RUN (OBS: Por enquanto, isto é apenas suposto, não digite nenhum comando), em seguida pressionamos RETURN(CR)■ .

E todas as expressões do programa serão executadas uma após a outra. Neste exemplo são as duas expressões PRINT. Quando você digitar RUN e pressionar RETURN (CR)■ ;é enviado ao computador um comando para a linha com o número de mais baixo valor, e inicia-se o programa a partir desse ponto. São executadas as expressões em ordem numérica, não sendo considerada a ordem em que foram digitadas originalmente.

■ RETURN(CR)

VISUAL 3

Mostra um programa em ordem invertida, isto é, a última linha foi colocada em primeiro lugar, mas mesmo assim quando RUN é acionado...

■ RETURN(CR)

E a tecla ■ RETURN(CR) é pressionada...

■ RETURN(CR)

O computador executa as linhas 3, depois 4 e depois 5. Observe que a instrução PRINT sem nada a seguir imprime uma linha em branco.

VISUAL 4

■ RETURN(CR)

Responda às perguntas propostas.

VISUAL 5

Aqui está o mesmo programa anterior.

Você pode perceber que os números das linhas podem ser colocados em 1, 10, 100 e assim por diante. Os números não precisam conter intervalos iguais, mas cada número de linha precisa ter um valor inferior a 64.000. Para repetir, em qualquer ordem que você der entrada no números das linhas, o sistema operacional vai colocá-las em ordem ascendente e depois executá-las.

Vamos executar este programa:

■ RETURN(CR)

O equipamento imprimiu RUN

■ RETURN(CR)

Se quiséssemos executar o mesmo programa várias vezes, simplesmente digitamos RUN e,

■ RETURN(CR)

E veremos os mesmos resultados novamente. Podemos executar várias vezes o mesmo programa porque ele continua na memória independentemente do número de vezes que for executado. Mas, se o programa não estiver mais na tela :

■ RETURN (CR)

Para que possamos vê-lo novamente, utilizamos o comando LIST

■ RETURN(CR)

VISUAL 6

■ RETURN(CR)

Mostra um programa fora de ordem. O comando LIST pode ser útil quando queremos visualizar o programa na ordem numérica correta.

■ RETURN(CR)

■ RETURN(CR)

E então visualizamos o programa na ordem em que vai ser executado.

■ RETURN(CR)

VISUAL 7

■ RETURN(CR)

A palavra LIST acabou de ser impressa.

■ RETURN(CR)

Você agora está observando a listagem do programa da AULA 2, que está assistindo agora. Obviamente, o tamanho deste programa ultrapassa a capacidade de ser impresso por inteiro na tela. O comando LIST pode ser usado para mostrar partes de um programa ou uma linha de instrução específica.

■ RETURN(CR)

VISUAL 8

Mostra o programa anterior. Se você quiser ver apenas a linha 15 deve digitar LIST 15

■ RETURN(CR) ■ RETURN(CR)

E, aí está a linha 15

Vamos supor que você queira ver todas as instruções das linhas entre 10 e

■ RETURN(CR)

E aí estão nossas linhas. Mais adiante, falaremos de novo sobre o comando LIST.

■ RETURN(CR)

A primeira parte da aula 2 está completa.

■ RETURN(CR)

Leia o Caderno de Exercícios e pratique.

■ RETURN(CR)

Desligue o equipamento. Quando estiver pronto, ligue-o novamente e escolha o número correspondente à AULA 2 PARTE 2

.....

Na primeira parte desta aula você aprendeu como deve ser digitado um programa. Agora vamos aprender como corrigir erros no programa e, como eliminar um programa da memória de seu computador.

■ RETURN (CR)

Vamos dizer que você acabou de introduzir um programa como este, e não percebeu que a instrução PRINT está escrita incorretamente na linha 30.

■ RETURN(CR)

Aparece o comando RUN

■ RETURN(CR)

Linhas 10 e 20 são executadas, porém o programa não continua porque o computador não compreende o significado da palavra PIRNT

Você receberá uma mensagem de erro com o número de linha onde isto ocorreu. Aqui está uma maneira de corrigir isto.

RETURN(CR)

E a linha 30 desaparecerá do programa. Para provar sua eliminação:

■ RETURN(CR)

■ RETURN(CR)

Aparece a palavra LIST na tela

RETURN (CR)

E, como pode ser observado não hã linha 30 na versão nova. Porém suponhamos que a linha 30 seja necessária para a execução do programa, e não podemos eliminã-la.

RETURN (CR)

VISUAL 2

Vamos corrigir o erro da linha 30

■ RETURN(CR)

Simplesmente introduzimos o número da linha e a seguir a instrução soletrada de modo correto, PRINT,

■ RETURN(CR)

E linha 30 está substituída.

Digitamos LIST

■ RETURN(CR)

■ RETURN(CR)

E aí está nossa nova linha 30 que substituiu a linha 30 defeituosa.

■ RETURN(CR)

Para ver o programa ser executado introduzimos RUN

■ RETURN(CR)

■ RETURN(CR)

Aí está o resultado.

VISUAL 3

OBS: Lembre-se que você também pode corrigir um erro, usando a tecla de retrocesso, flecha apontada para a esquerda, se a tecla RETURN(CR) não tiver sido pressionada..

VISUAL 4

Suponhamos que queremos eliminar um programa inteiro, para que não haja - possibilidade de que uma linha do programa anterior, seja incluída no novo programa. Elimina-se o programa anterior da memória, digitando-se o comando NEW

■ RETURN(CR)

NEW

■ RETURN(CR)

■ RETURN(CR)

Assim o programa todo será eliminado da memória. Para provar sua eliminação,

LIST

■ RETURN(CR)

E, não há listagem na tela, porque não há programa nenhum na memória.

■ RETURN(CR)

VISUAL 5

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 6

AULA 3 PARTE 1

Como ilustração você pode considerar a memória de um computador, como um conjunto de caixas.

■ RETURN(CR)

VISUAL 1

Como é visto aqui, cada uma dessas caixas pode ser designada por um nome.

■ RETURN(CR)

Daremos nomes para algumas delas; um nome que consiste de uma LETRA. Podemos colocar um número dentro de cada caixa.

■ RETURN(CR)

Assim, um número foi colocado dentro de uma caixa, cuja referência é uma letra. Por exemplo, para utilizarmos o número 20 na execução de funções aritméticas, nós endereçaremos o número pelo seu nome, isto é A.

Podemos colocar um número em qualquer caixa se dermos a seguinte entrada:

■ RETURN(CR)

LET W = 24

■ RETURN(CR)

Veremos então a caixa com o endereço W que conterá o valor 24, ou podemos omitir a instrução LET

■ RETURN(CR)

E apenas dar entrada a Z = 2

■ RETURN(CR)

E vemos a caixa Z com o valor 2 como conteúdo

■ RETURN(CR)

VISUAL 2

Variáveis, podem ter como conteúdo, a idade média de um grupo, acumular o número de vezes em que houve erros, armazenar o valor de uma área, e assim por diante. Você verá como esta função é útil quando queremos tratar com números grandes, ou com números cujo valor exato não é necessário saber, porém seu armazenamento na memória é necessário para executar funções matemáticas futuras.

■ RETURN(CR)

VISUAL 3

Como é visto aqui, você pode utilizar como variável qualquer letra de A a Z

■ RETURN(CR)

Ou, qualquer combinação de 2 letras de AA até ZZ

■ RETURN(CR)

Ou, ainda qualquer letra seguida por um número de 0 a 9

■ RETURN(CR)

VISUAL 4

Para termos o conteúdo de uma variável usamos a instrução PRINT

Vamos executar este programa

■ RETURN(CR)

■ RETURN(CR)

Uma caixa está com o endereço X e, o número 6 será colocado como seu conteúdo. O conteúdo de X é impresso na tela, e o resultado é obviamente 6.

A instrução LET tanto como PRINT, pode ser usada como comando, isto é sem número de linha ou mesmo sem qualquer programa.

■ RETURN(CR)

Entra-se com Y = 50

■ RETURN(CR)

■ RETURN(CR)

■ RETURN(CR)

E aí está

O comando PRINT pode ser de utilidade às vezes, depois que o programa foi executado, quando você quiser saber o conteúdo de uma variável.

Se você utilizar o comando PRINT para saber o conteúdo de uma variável ainda não utilizada,

■ RETURN(CR)

A variável será inicializada com o valor zero.

■ RETURN(CR)

Não se esqueça de que qualquer variável, sempre que for usada pela primeira vez terá como valor zero.

Agora, Responda às perguntas propostas

■ RETURN(CR)

Como variáveis representam os valores de seus conteúdos, elas podem substituir números reais em Aritmética e outras Operações.

Temos, como exemplo, a impressão da variável X + 5

■ RETURN(CR)

O conteúdo de X que é 6, é somado com 5 resultando 11. Percebe-se que o valor de X não foi alterado. Se você quiser aumentar o conteúdo de X digamos de 8,

■ RETURN(CR)

Damos como entrada: X + 8

■ RETURN(CR)

O valor de X fica então modificado, e o novo valor que é 14 vai ser o conteúdo. Supondo que queremos armazenar a soma de X + Y dentro de uma nova variável,

■ RETURN(CR)

Damos entrada: A = X + Y

■ RETURN(CR)

A é então inicializado e utilizado para armazenar a soma de $X + Y$ ou seja 64. Observe que os valores de X e Y não sofreram nenhuma alteração e, finalmente colocamos o valor de A em Z

■ RETURN (CR)

LET Z = A

■ RETURN(CR)

Z então armazena um novo valor. O valor anterior de Z é substituído.

■ RETURN(CR)

VISUAL 5

Responda às perguntas propostas

VISUAL 6

A primeira parte da AULA 3 está terminada. Leia o Caderno de Exercícios e pratique.

■ RETURN(CR)

Desligue o computador. Ao ligá-lo digite o número correspondente à AULA 3 PARTE 2

.....

AULA 3 PARTE 2

Nesta aula, você vai aprender como utilizar a instrução INPUT.

■ RETURN(CR)

VISUAL 1

■ RETURN(CR)

Se este programa fosse executado, você encontraria a área de um quadrado, cujo lado é igual a 10, por causa da instrução LET da linha 20.

■ RETURN(CR)

Se você precisasse encontrar a área de um quadrado cujo lado tivesse 12, 15 ou 500; você precisaria reescrever a linha 20 de novo, e de novo, e de novo, para cada novo valor do lado do quadrado.

Aqui está uma solução melhor.

■ RETURN(CR)

Observe, é o mesmo programa, exceto pela linha 20 que foi modificada.

■ RETURN(CR)

A linha 20 agora, é uma instrução INPUT, vamos entrar com RUN

■ RETURN(CR)

■ RETURN(CR)

O programa temporariamente está na linha 20, na instrução INPUT (?), o computador espera que se entre com um número e ,

■ RETURN(CR)

Quando executamos essas ações, a variável S será igual ao número que dermos como entrada. Vamos entrar com um número,

■ RETURN(CR)

■ RETURN(CR)

O computador armazena o número 12, fazendo com que $S = 12$, e faz A igual a $S * S$ e imprime 144 para nós (A)

■ RETURN(CR) Resposta às perguntas propostas

VISUAL 3

O programa observado, está simulando uma execução. A linha 10 avisa que, - o computador espera que você entre com 2 números separados por uma vírgula

■ RETURN(CR)

5 , 3

■ RETURN(CR)

E sua resposta aparece

■ RETURN(CR)

RUN

■ RETURN(CR)

Desta vez vamos deliberadamente esquecer de entrar com o segundo número.

■ RETURN(CR)

Aparece na tela dois pontos de interrogação ??

Significam que o computador está lhe pedindo a entrada do segundo número.

Entramos com ele,

■ RETURN(CR)

■ RETURN(CR)

Vamos executá-lo pela terceira vez:

■ RETURN(CR)

■ RETURN(CR)

■ RETURN(CR)

Agora tentamos entrar com 2 números

■ RETURN(CR)

Porém erramos, e um dos números é uma letra. Vamos ver o que ocorre quando

■ RETURN(CR)

Seu computador fornecerá uma mensagem de REENTRE. Entre com seu dado novamente, e agora correto.

■ RETURN(CR)

■ RETURN(CR)

E sua resposta será dada.

Uma mensagem REENTRE, sempre aparece quando você procura entrar com qualquer símbolo ou letra em uma variável, que sempre aceita números.

VISUAL 4

VISUAL 5

Muitas vezes esses resultados são causados pelo usuário, que se esquece do INPUT, ou quantos INPUT's são necessários. Mesmo se o usuário for você, pode ocorrer esquecimento. Nós vamos mostrar-lhe um exemplo de execução de programa. Este programa possui uma instrução extra em seu conteúdo. A instrução HOME na linha 10.

Essa instrução limpa a tela por compélto, e imprime o próximo PRINT no canto superior esquerdo. Pode ser utilizada como instrução ou comando. Vamos presumir que faz um ano que não executamos este programa.

VISUAL 6

E, obtive uma resposta. Porém o que significa 5 e 6 ? O que significa a resposta?

RETURN(CR)

VISUAL 7

RETURN(CR)

Observe a instrução HOME para limpar a tela. Existem várias instruções - PRINT sem nada escrito em seguida, que servem para providenciar espaço - entre as linhas.

Linhas 50 a 80, dizem ao usuário exatamente o que deverá ser feito.

Linhas 100 a 120, explicam ao usuário como devem ser dadas as entradas de seus valores. Linha 50 imprimirá a resposta. Vamos ver esse programa .

RETURN(CR)

VISUAL 8

A tela está limpa, e as mensagens nos dizem que o computador está esperando as entradas referentes aos dois lados. Vamos fazer o que o computador pede entrando com os valores.

RETURN(CR)

E nós recebemos a resposta com seu significado..

RETURN(CR)

VISUAL 9

Responda às perguntas propostas.

VISUAL 10

Este é o final. Leia o Caderno de Exercícios e pratique. Desligue o computador

VISUAL 1

Nesta aula você aprenderá instruções que distinguem os computadores das calculadoras, realizando Decisões.

E, para iniciar, a primeira Instrução que vamos estudar é o GOTO.

Observe o programa na tela.

Por mais elementar que possa parecer, o controle de um programa, é sempre de cima para baixo e em ordem numérica. Um comando foi digitado aqui, GOTO 30.

■ RETURN(CR)

Quando a tecla acima é pressionada, o computador vai para a linha 30 e depois continua para baixo, isto é linha 40.

Observe que a instrução PRINT na linha 30 foi executada, depois o programa foi para a linha 40 e terminou.

■ RETURN(CR)

VISUAL 2

A instrução GOTO, pode ser parte de um programa, como é mostrado na tela.

A linha 10 será executada, depois a 20, e a instrução GOTO mandará o controlador para a linha 50, terminando assim a execução do programa.

■ RETURN(CR)

Observe que as linhas 30 e 40, nunca serão executadas.

■ RETURN(CR)

VISUAL 3

Perguntas.

VISUAL 4

A instrução GOTO, pode fazer um programa entrar em LOOP; ação de sair de uma linha e ir para outra e, retornar a linha de origem repetindo a sequência várias vezes.

■ RETURN(CR)

Aqui, a linha 10 segue para a linha 20, de 20 para 30 e, então a instrução GOTO na linha 30 envia o controle de volta para a linha 10 e, o ciclo se repete.

Observe que a linha 40 nunca será executada.

Agora vamos demonstrar para você; UMA FRAÇÃO DE TEMPO DE EXECUÇÃO DESTE PROGRAMA.

■ RETURN(CR)

Primeiro entramos com RUN

■ RETURN(CR)

O programa irá imprimir para sempre se você permitir. A execução deste pro-

grama é um LOOP infinito, porque nunca terminará.

■ RETURN(CR)

VISUAL 5

Mostra outro programa em LOOP infinito.

Este programa continuará imprimindo qualquer número dado como entrada, infinitamente.

Quando um LOOP for iniciado, você precisa fazer com que ele pare de alguma maneira. As vezes, um erro provocará sua entrada em LOOP infinito e, será necessário que você pare o programa.

Para fazer isso, pressione a tecla CTRL e o caracter C simultaneamente.

■ RETURN(CR)

CTRL-C, gera uma mensagem para você e, para a execução do programa mostrando na tela a mensagem PAROU EM :, com o número da linha onde o programa parou sua execução. Apesar de que CTRL-C pode parar um LOOP infinito, pode ser uma idéia melhor interceptar o LOOP sem interromper o programa.

■ RETURN(CR) Responda às perguntas propostas

VISUAL 6

Nos mostra, como uma instrução IF - THEN funciona.

No momento em que o computador lê a instrução IF da linha 30 e, a condição dada for satisfeita, tudo que estiver após a instrução THEN, será executado. Se a condição não for satisfeita, o restante da linha, isto é, o que segue após a instrução THEN será ignorado, e o controle passará a executar a linha seguinte, no caso a linha 40.

■ RETURN(CR)

RUN Entramos com RUN

■ RETURN(CR)

? A linha 10 espera que entremos com INPUT em A.

■ RETURN(CR)

? 3 Foi dada a entrada.

■ RETURN(CR)

3 A linha 20 foi executada, isto é, A foi impresso.

?

■ RETURN(CR)

? 800 O controle seguiu para a linha 40, e retornou para a linha 10 e, aguarda um novo INPUT

■ RETURN(CR)

800

?

A linha 20 imprimiu A como anteriormente, e o computador aguarda um novo INPUT

■ RETURN(CR)

? 0

Desta vez o INPUT A = 0

■ RETURN(CR)

0

PAROU EM 30

A linha 20 imprimiu A. Como o INPUT A = 0, houve comparação de A na linha 30 e, foi impressa a mensagem onde ocorreu a parada.

A certamente foi igual a 0. Como são iguais a linha 30 foi totalmente executada, isto é, a instrução THEN foi executada, terminando assim o programa.

■ RETURN(CR)

VISUAL 7

Perguntas

VISUAL 8

Observe o programa.

Qualquer uma das três palavras, poderá ser impressa na tela, dependendo do número utilizado no INPUT A.

Observe que, se o INPUT A não for igual a 1, 2 e 3, o controle passa diretamente para a instrução END na linha 50 e, não imprime nada.

Vamos experimentar a execução do programa.

■ RETURN(CR)

RUN

Nós entramos com RUN. O computador espera então que coloquemos um número.

■ RETURN(CR)

Introduzimos o número 3 na Variável A, cujo valor não é igual a 1 ou 2, portanto as linhas 20 e 30 são ignoradas, sendo que a linha 40 é executada imprimindo TALVEZ. Vamos executá-lo novamente.

■ RETURN(CR)

Introduzimos um outro número.

■ RETURN(CR)

Introduzimos o número 2 na Variável A, cujo valor não é igual a 1 ou 3, portanto as linhas 20 e 40 são ignoradas, sendo que a linha 30 é executada imprimindo NÃO. Vamos executá-lo novamente.

■ RETURN(CR)

Entramos com RUN.

Introduzimos o número 1. Portanto SIM foi impresso, e as linhas 30 e 40 serão ignoradas. Mais exemplos.

RUN Entramos com RUN , e outro número.

Agora a Variável A não é igual a 1,2 ou 3. Portanto desta vez as linhas 20, 30 e 40 foram ignoradas.e o programa termina sem nada imprimir.

VISUAL 9

Perguntas.

A primeira parte da Aula 4 está completa. Leia o Caderno de Exercícios e pratique. Desligue seu computador. Ao ligá-lo digite o número correspondente à Aula 4 Parte 2.

0001 0002 0003 0004 0005 0006 0007 0008 0009 0010 0011 0012 0013 0014 0015 0016 0017 0018 0019 0020 0021 0022 0023 0024 0025 0026 0027 0028 0029 0030 0031 0032 0033 0034 0035 0036 0037 0038 0039 0040 0041 0042 0043 0044 0045 0046 0047 0048 0049 0050 0051 0052 0053 0054 0055 0056 0057 0058 0059 0060 0061 0062 0063 0064 0065 0066 0067 0068 0069 0070 0071 0072 0073 0074 0075 0076 0077 0078 0079 0080 0081 0082 0083 0084 0085 0086 0087 0088 0089 0090 0091 0092 0093 0094 0095 0096 0097 0098 0099 0100 0101 0102 0103 0104 0105 0106 0107 0108 0109 0110 0111 0112 0113 0114 0115 0116 0117 0118 0119 0120 0121 0122 0123 0124 0125 0126 0127 0128 0129 0130 0131 0132 0133 0134 0135 0136 0137 0138 0139 0140 0141 0142 0143 0144 0145 0146 0147 0148 0149 0150 0151 0152 0153 0154 0155 0156 0157 0158 0159 0160 0161 0162 0163 0164 0165 0166 0167 0168 0169 0170 0171 0172 0173 0174 0175 0176 0177 0178 0179 0180 0181 0182 0183 0184 0185 0186 0187 0188 0189 0190 0191 0192 0193 0194 0195 0196 0197 0198 0199 0200 0201 0202 0203 0204 0205 0206 0207 0208 0209 0210 0211 0212 0213 0214 0215 0216 0217 0218 0219 0220 0221 0222 0223 0224 0225 0226 0227 0228 0229 0230 0231 0232 0233 0234 0235 0236 0237 0238 0239 0240 0241 0242 0243 0244 0245 0246 0247 0248 0249 0250 0251 0252 0253 0254 0255 0256 0257 0258 0259 0260 0261 0262 0263 0264 0265 0266 0267 0268 0269 0270 0271 0272 0273 0274 0275 0276 0277 0278 0279 0280 0281 0282 0283 0284 0285 0286 0287 0288 0289 0290 0291 0292 0293 0294 0295 0296 0297 0298 0299 0300 0301 0302 0303 0304 0305 0306 0307 0308 0309 0310 0311 0312 0313 0314 0315 0316 0317 0318 0319 0320 0321 0322 0323 0324 0325 0326 0327 0328 0329 0330 0331 0332 0333 0334 0335 0336 0337 0338 0339 0340 0341 0342 0343 0344 0345 0346 0347 0348 0349 0350 0351 0352 0353 0354 0355 0356 0357 0358 0359 0360 0361 0362 0363 0364 0365 0366 0367 0368 0369 0370 0371 0372 0373 0374 0375 0376 0377 0378 0379 0380 0381 0382 0383 0384 0385 0386 0387 0388 0389 0390 0391 0392 0393 0394 0395 0396 0397 0398 0399 0400 0401 0402 0403 0404 0405 0406 0407 0408 0409 0410 0411 0412 0413 0414 0415 0416 0417 0418 0419 0420 0421 0422 0423 0424 0425 0426 0427 0428 0429 0430 0431 0432 0433 0434 0435 0436 0437 0438 0439 0440 0441 0442 0443 0444 0445 0446 0447 0448 0449 0450 0451 0452 0453 0454 0455 0456 0457 0458 0459 0460 0461 0462 0463 0464 0465 0466 0467 0468 0469 0470 0471 0472 0473 0474 0475 0476 0477 0478 0479 0480 0481 0482 0483 0484 0485 0486 0487 0488 0489 0490 0491 0492 0493 0494 0495 0496 0497 0498 0499 0500 0501 0502 0503 0504 0505 0506 0507 0508 0509 0510 0511 0512 0513 0514 0515 0516 0517 0518 0519 0520 0521 0522 0523 0524 0525 0526 0527 0528 0529 0530 0531 0532 0533 0534 0535 0536 0537 0538 0539 0540 0541 0542 0543 0544 0545 0546 0547 0548 0549 0550 0551 0552 0553 0554 0555 0556 0557 0558 0559 0560 0561 0562 0563 0564 0565 0566 0567 0568 0569 0570 0571 0572 0573 0574 0575 0576 0577 0578 0579 0580 0581 0582 0583 0584 0585 0586 0587 0588 0589 0590 0591 0592 0593 0594 0595 0596 0597 0598 0599 0600 0601 0602 0603 0604 0605 0606 0607 0608 0609 0610 0611 0612 0613 0614 0615 0616 0617 0618 0619 0620 0621 0622 0623 0624 0625 0626 0627 0628 0629 0630 0631 0632 0633 0634 0635 0636 0637 0638 0639 0640 0641 0642 0643 0644 0645 0646 0647 0648 0649 0650 0651 0652 0653 0654 0655 0656 0657 0658 0659 0660 0661 0662 0663 0664 0665 0666 0667 0668 0669 0670 0671 0672 0673 0674 0675 0676 0677 0678 0679 0680 0681 0682 0683 0684 0685 0686 0687 0688 0689 0690 0691 0692 0693 0694 0695 0696 0697 0698 0699 0700 0701 0702 0703 0704 0705 0706 0707 0708 0709 0710 0711 0712 0713 0714 0715 0716 0717 0718 0719 0720 0721 0722 0723 0724 0725 0726 0727 0728 0729 0730 0731 0732 0733 0734 0735 0736 0737 0738 0739 0740 0741 0742 0743 0744 0745 0746 0747 0748 0749 0750 0751 0752 0753 0754 0755 0756 0757 0758 0759 0760 0761 0762 0763 0764 0765 0766 0767 0768 0769 0770 0771 0772 0773 0774 0775 0776 0777 0778 0779 0780 0781 0782 0783 0784 0785 0786 0787 0788 0789 0790 0791 0792 0793 0794 0795 0796 0797 0798 0799 0800 0801 0802 0803 0804 0805 0806 0807 0808 0809 0810 0811 0812 0813 0814 0815 0816 0817 0818 0819

VISUAL 1

Na Parte 1, você aprendeu a instrução GOTO.

GOTO não é útil somente na formação de LOOP's, podendo ser usado em conjunto com outras instruções, tais como IF - THEN.

Vamos executar esse programa.

Entramos com o número 1

A linha 20 comparou X com o número 1, satisfazendo a condição de igualdade, executando assim as linhas 50,60 e 70, imprimindo o valor de X, FIM e parando a execução do programa.

Vamos executá-lo novamente.

A linha 20 encontra o valor de X não igual a 1, portanto executa a linha 30. X é encontrado e é igual a 2. Então o controle envia o comando para a linha 60 imprimindo a palavra FIM, e parando a execução do programa; vamos executá-lo novamente.

Desta vez introduzimos como INPUT o número 3.

■ RETURN(CR)

Nenhuma das condições foi satisfeita, portanto o controlador foi para a linha 10, ignorando as linhas 20 e 30 e, executando a seguir a linha 40 onde se encontra a instrução STOP.

O que foi impresso, se você lembrar foi o número da linha que você parou fornecido sempre em conjunto com a instrução STOP.

■ RETURN(CR)

VISUAL 2

Perguntas.

VISUAL 3

Até agora, tratamos apenas com o símbolo de igualdade. Há outros símbolos que podemos usar.

■ RETURN(CR)

O símbolo de igual(=), é eficaz enquanto trabalhamos com dois valores que são idênticos, porém se um valor for maior que o outro temos de ter uma maneira de mostrar quem é o maior.

■ RETURN(CR)

Na medida que se aumenta em valor, o símbolo de = separa-se e abre pelo lado esquerdo, significando que Z precisa diminuir para ser = a N.

O símbolo no meio é chamado de MAIOR QUE.

■ RETURN(CR)

No sentido inverso, se N é maior, o símbolo abrirá novamente no sentido do número de maior valor, significando que Z precisa aumentar seu valor para ser = a N.

■ RETURN(CR)

O símbolo do meio é chamado de MENOR QUE.

■ RETURN(CR)

VISUAL 4

Resumindo:

X é maior que Y, ou X tem de diminuir seu valor para ser = a Y.

■ RETURN(CR)

X é menor que Y, ou X tem de aumentar seu valor para ser = a Y.

■ RETURN(CR)

VISUAL 5

Observe o programa.

A linha 30 deve ser lida da seguinte maneira:

Se nosso INPUT para X, é maior do que nosso INPUT para M, então pare o pro-

grama. Agora responda as perguntas.

■ RETURN(CR)

VISUAL 6

Esses símbolos podem ter várias combinações em uma instrução.

Aqui, o símbolo MENOR QUE, é seguido pelo símbolo de igual, isto significa que X é MENOR OU IGUAL A Y.

■ RETURN(CR) ■ RETURN(CR) ■ RETURN(CR)

VISUAL 7

Mostra todas as combinações possíveis entre os sinais.

Perguntas.

■ RETURN(CR)

VISUAL 8

Este é o final da Aula 4. Leia o Caderno de Exercícios e pratique. Desligue o computador. Ao ligá-lo digite o número correspondente ao TESTE 4.

.....

VISUAL 1

Na última aula, você aprendeu a utilizar as instruções, GOTO, IF - THEN. Nessa aula, você aprenderá como utilizar contadores em programas, para, controlar LOOP's; e como utilizar a instrução RUN.

Nós já vimos na aula anterior, como LOOP's infinitos podem continuar para sempre, e precisam ser parados com o comando CTRL-C.

LOOP's infinitos, são praticamente inúteis para o nosso propósito.

Porém um LOOP finito, isto é, que pára por si mesmo, sem dúvida será útil.

Nós conhecemos uma maneira para criar um LOOP finito, com a utilização de uma Variável de valor = 0(ZERO) em conjunto com uma instrução IF - THEN.

Existe um outro meio, contar quantas vezes ocorre um LOOP e depois parar quando o limite pré-determinado for alcançado.

Antes de examinar como um computador realiza a contagem, vamos examinar, como as pessoas contam.

■ RETURN(CR)

Quando você conta, você inicia sua contagem do 0(ZERO).

Adicionamos um(1) ao valor ZERO(0) e temos 1.

Adicionamos 1 a soma anterior e teremos 2.

Adicionamos mais 1 e teremos 3, e assim por diante.

Em termos explícitos de programação, nós podemos estabelecer Zero(0) para a Variável A com a instrução LET A = 0

■ RETURN(CR)

Você vai lembrar da Aula 3, onde foi visto como podemos aumentar o valor de uma Variável, permitindo assim que a Variável A possa ser somada a qualquer número. EX: $A = A + 1$.

Então cada vez que usarmos a instrução LET $A = A + 1$,

■ RETURN(CR)

A Variável A, terá seu valor acrescido de 1 a cada execução do LOOP.

Agora vamos fazer esse programa contar de 0 até 6.

Primeiro podemos adicionar ao programa a linha número 5, para permitir que $A = 0$.

■ RETURN(CR)

Lembre-se que na realidade não precisamos fazer isto, pois todas as Variáveis mencionadas pela primeira vez no programa, são condicionadas a assumirem o valor 0(ZERO).

Porém vamos utilizar esta instrução de qualquer maneira.

Vamos pedir uma listagem para vermos o que temos até agora.

■ RETURN(CR)

Como pode ser visto, temos um programa que começa inicializando a Variável A com o valor 0(ZERO).

Agora logo após a instrução PRINT, nós teremos de incrementar a Variável A de 1.

Então, precisamos de uma linha de instrução após a linha 10, no caso a linha 13 a qual conterá a instrução $LET A = A + 1$.

■ RETURN(CR)

Pedimos uma nova listagem, com a inclusão da linha 13.

■ RETURN(CR)

Portanto depois da primeira instrução PRINT, teremos um novo valor para A, que será igual a 1.

A linha 20 mandará o controle para a linha 10.

Fará outro PRINT da Variável A, desta vez com o valor 1 e, depois aumentará para 2.

Mais um LOOP e o valor será de 3, e assim por diante.

Então, como pode ser observado o programa está contando o número de vezes que o LOOP é executado.

Porém, apesar de que A está sendo acumulado na contagem dos LOOP's, o LOOP, ainda está infinito; precisamos pará-lo após o contador chegar ao valor 6.

Então, colocaremos um fim condicional ao programa.

Para isso é necessário uma outra linha depois da linha 13. Vamos dizer linha 16.

■ RETURN(CR)

IF A = 6 THEN END, ou seja se A for igual a 6 então FIM.

Pedimos uma nova listagem, com a inclusão da linha 16.

■ RETURN(CR)

Agora cada vez que ocorrer um PRINT do valor de A, o contador A é incrementado de 1, e A é comparado ao valor 6. Se for igual a 6, então o programa, termina. Se não for a linha 20 manda o controle para a linha 10, para ser executado mais um LOOP.

Vamos executar este programa.

■ RETURN(CR)

Aí está nossa rotina de contagem de 1 a 6.

OBS: Contadores de programas, não são utilizados apenas para limitar LOOP's dentro de um programa. Existem outras utilidades.

■ RETURN(CR)

VISUAL 2

PERGUNTAS:

VISUAL 3

Mostra um programa que contém um contador para verificar o número de erros. N, é um número a ser adivinhado, como estabelecido na linha 10.

A linha 35 aceita a entrada de sua adivinhação. Se a sua adivinhação não for correta, a linha 40 incrementa o contador de erros e, você recebe uma mensagem de erro.

A linha 50 manda o controle para a linha 35, para que seja dada outra entrada.

Percebe-se que o valor de N, nunca é modificado devido ao LOOP, apenas o contador de erros é modificado.

Se a sua adivinhação estiver correta, as linhas 40,45 e 50 serão ignoradas, porque a condição IF não foi satisfeita.

Então, você receberá do computador uma mensagem de congratulações.

Observe que a linha 80, imprime o acúmulo de todos os erros realizados por você.

Vamos executar o programa:

■ RETURN(CR)

RUN

■ RETURN(CR)

Escolha um número entre 1 e 10.

■ RETURN(CR)

Número errado. Tente novamente.

■ RETURN(CR)

Errou de novo. Mais uma tentativa.

■ RETURN(CR)

Você conseguiu!!!!!! E também temos o número de vezes que você errou. Perguntas.

VISUAL 4

Nossos programas agora, podem ser longos e mais completos. Para conseguir reconhecer partes de um programa, nós utilizaremos a instrução REM.

Observe o programa.

■ RETURN(CR)

Quando o computador encontra uma instrução REM, ele ignora tudo que estiver escrito em seguida. Vamos examinar a execução do programa.

■ RETURN(CR)

Realmente a linha 10, não provocou nada, a não ser etiquetar o programa, quando olhamos a listagem.

■ RETURN(CR)

VISUAL 5

Observe o programa.

Novamente durante a execução as linhas 10 e 20 serão ignoradas e, apenas as linhas 30 e 40 serão executadas.

Quando fazemos a listagem, todas as informações inclusas em conjunto com a instrução REM estarão presentes.

Você descobrirá como esta instrução pode ser de extrema importância quando escrever programas extensos.

Você deve saber que para conservar na memória do computador, depois de ter verificado seu programa, as expressões em conjunto com as instruções REM, podem ser eliminadas do programa: ■ RETURN(CR)

Porém, quando possível a não eliminação destas instruções facilitam seu trabalho.

■ RETURN(CR)

Agora vamos ver a listagem do programa.

VISUAL 6

Perguntas.

VISUAL 7

Este é o final da Aula 5 Parte 1. Leia o Caderno de Exercícios e pratique. Desligue o computador. Ao ligá-lo, digite o número correspondente à Aula 5 Parte 2.

.....

AULA 5 PARTE 2

VISUAL 1

Nesta aula, você vai aprender algumas rotinas que o interpretador de BASIC pode realizar para você, automaticamente.

Essas rotinas compõem a Biblioteca de Funções.

■ RETURN(CR)

A Função INTEIRO, converte qualquer número em um número inteiro.

■ RETURN(CR)

A Função ABSoluta, converte qualquer número em um número positivo.

■ RETURN(CR)

A Função RaNDômica, escolhe qualquer número aleatoriamente.

Vamos deixar bem claro que, qualquer uma das funções aqui descritas, pode

ser escrita por você ou pelo computador, como parte de um programa em linguagem BASIC.

Estas funções são utilizadas com tanta frequência que os fabricantes de computadores, decidiram incluí-las no Interpretador BASIC. É por esta razão que são denominadas Funções Bibliotecárias.

As funções fazem parte integral do Interpretador BASIC.

■ RETURN(CR)

VISUAL 2

Um número INTEIRO, é um número sem nenhum valor decimal, ou frações a ele ligado.

■ RETURN(CR)

Como mostra o exemplo na tela, 3 é um número inteiro; já 4.98 não o é. Para criar um número inteiro, o computador elimina tudo que estiver a direita do ponto decimal.

■ RETURN(CR)

Como pode ser visto, utilizamos o comando PRINT para imprimir o inteiro de 4.98 e, o computador imprime o número 4 para nós. O número 4 é o inteiro de 4.98.

Para fazer-mos uso de qualquer uma destas funções, usa-se sempre o mesmo formato em BASIC.

Digite a função, e em seguida um número ou variável entre parentesis.

A Função INTEIRO, pode às vezes ser de grande valor quando queremos que o computador execute algo específico, cada outra vez ou a cada tres vezes.

■ RETURN(CR)

VISUAL 3

O contador X, inicia o valor incrementado de 1(um) e, está contando LOOP's até 8.

Se você observar a coluna de resultados, verá apenas dois números inteiros cada vez que o valor de X é divisível por tres e, o resultado for inteiro. Observe que são duas as únicas vezes que o inteiro do resultado é igual ao próprio resultado.

■ RETURN(CR)

Então, se você quisesse que seu programa executasse uma certa rotina a cada tres vezes que ele passa pelo LOOP, a única coisa a ser feita é comparar o contador dividindo-o por 3, junto com o inteiro do contador dividido por 3 como segue na tela:

■ RETURN(CR)

Uma condição de igualdade, existe apenas uma a cada tres vezes.

■ RETURN(CR)

VISUAL 4

Imagine um campo de futebol, partindo do meio de campo (PONTO ZERO), uma flecha mostra o ataque, tendo conseguido um avanço de 10 metros(+10). A outra mostra um contra-ataque onde nosso time perdeu 10 metros(-10). O sinal do número (+ ou -), demonstra o sentido em que a flecha se dirige. O número 10 sem sinal, (+ ou -), é chamado de Valor ABSoluto.

■ RETURN(CR)

Aqui o valor de -10 ou +10 é, simplesmente 10. A Função ABSoluta, elimina o sinal de qualquer número.

■ RETURN(CR)

VISUAL 5

Para conseguirmos um valor ABSoluto de qualquer número, digita-se ABS, com o número ou variável entre parentesis.

■ RETURN(CR) ■ RETURN(CR)

VISUAL 6

Perguntas.

VISUAL 7

Agora, chegamos na Função RaNDômica ou "RANDOM".

Números randômicos são escolhidos aleatoriamente. Exemplos:

■ RETURN(CR)

Assinalamos uma variável para ser igual a RND(1) e esta será modificada em qualquer número de 0 até o último número antes do valor de 1. Se quisermos ver o números, introduza apenas PRINT e a variável.

Vamos observar o exemplo a seguir.

■ RETURN(CR)

Nós colocamos um LOOP para imprimir 5 números randômicos. Note-se que todos são decimais, maiores que 0 e menores que 1. Lógicamente pode-se imprimir números randômicos sem variáveis.

■ RETURN(CR)

O que deve ser lembrado é que, o número randômico pode ser de 0 até um valor menor que 1. Para resumir aqui estão algumas perguntas.

■ RETURN(CR)

Não há problema em usar números decimais, porém nós podemos usar números maiores do que 1.

Para consegui-los, apenas multiplique seu número randômico pela quantidade que você deseja expandir.

■ RETURN(CR)

Aqui cada número é multiplicado por 50, resultando números que quase atin-
gem a 50. Agora vamos nos livrar das decimais.

Para conseguirmos isto, utilizaremos a função INT, que acabamos de aprender

■ RETURN(CR)

Agora temos números inteiros.

Mas espere,

■ RETURN(CR)

E se quisermos um número até e incluindo o 50?. Nenhum número decimal mul-
tiplicado por 50, será igual a 50, então teremos de somar 1.

■ RETURN(CR)

Este é o método utilizado para conseguirmos um número entre 1 e qualquer
outro números que desejarmos, números randômicos inteiros.

■ RETURN(CR)

VISUAL 8

Perguntas.

VISUAL 9

RESUMINDO:

Para conseguirmos um número randômico entre 1 e 50, entre com RND(1) e mul-
tiplique por 50, retire a parte decimal com a função INT e some 1.

Isto é frequentemente abreviado da seguinte maneira, como é mostrado a se-
guir na tela.

■ RETURN(CR)

Você deverá lembrar-se que, em qualquer operação matemática, quando paren-
thesis estão dentro de parentesis,

■ RETURN(CR)

O par interno é executado primeiro.

■ RETURN(CR)

VISUAL 10

Vamos agora cobrir mais uma Função que é RAIZ QUADRADA.

Digitando-se SQR, seguido por um número ou parentesis, conseguimos um núme-
ro que, quando multiplicado por si mesmo será igual ao número que especifi-
camos.

A tela nos mostra o comando PRINT e os resultados que o computador fornece.

■ RETURN(CR) ■ RETURN(CR)

VISUAL 11

Perguntas.

VISUAL 12

Para todos aqueles que se consideram matemáticos, aqui está uma lista de outras funções que nos podem ser úteis:

SENO, COSSENO, TANGENTE, ARCOTANGENTE em Radianos, bem como LOGARÍTIMOS.

Observe que todas utilizam o mesmo formato.

Função, seguida por um número ou uma Variável entre parentesis.

■ RETURN(CR)

VISUAL 13

Este é o final da Aula 5. Leia o Caderno de Exercícios e pratique. Desligue o computador. Ao ligá-lo, digite o número correspondente ao TESTE 5.

■ RETURN(CR)

■ RETURN(CR)

.....

AULA 6 PARTE 1

Nesta aula vamos discutir os comandos que controlam a tela.

VISUAL 1

Congratulações . Você está no meio do caminho deste curso. Você já possui um conhecimento razoável em BASIC. Nesta aula você vai aprender com que o computador, ao imprimir na tela apresente melhor aparência , isto é, que a impressão fique mais clara.

■ RETURN(CR)

VISUAL 2

Uma das primeiras coisas que desejamos fazer é limpar a tela. Como foi visto na aula anterior utilizamos a instrução HOME.

■ RETURN(CR)

VISUAL 3

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 4

Em nossa última aula também discutimos como evitar abarrotamentos (confusão) na tela. As linhas digitadas podem ser separadas com as instruções PRINT - como se vê na linha 20.

■ RETURN(CR)

Quando esse programa é executado, A e B são impressos com uma linha vazia intercalada.

■ RETURN(CR)

VISUAL 5

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 6

Quando uma vírgula aparece em conjunto com a instrução PRINT, o material em seguida da vírgula é colocado na próxima coluna da tela.

■ RETURN(CR)

Quando este programa é executado, 1 e 2 são separados em colunas .

■ RETURN(CR)

VISUAL 7

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 8

Quando você utiliza ponto e vírgula, nenhum espaço vai separar palavras ou

números.

■ RETURN(CR)

VISUAL 9

Quando a linha 30 deste programa for executada, as letras A e B serão impressas.

■ RETURN(CR)

Quando a linha 40 for executada os valores de A e B serão impressos.

■ RETURN(CR)

VISUAL 10

O ponto e vírgula, também avisa o computador para continuar imprimindo na mesma linha ao invés de descer para a próxima.

■ RETURN(CR)

VISUAL 11

Quando este programa é executado...

■ RETURN(CR)

O computador imprime para trás e para cima em linhas sucessivas. Quando o ponto e vírgula é adicionado, o computador vai imprimir assim...

■ RETURN(CR)

VISUAL 12

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 13

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 14

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 15

A próxima instrução de PRINT que vamos discutir é o comando TAB

■ RETURN(CR)

VISUAL 16

O comando TAB funciona da mesma maneira que um tabulador de máquina de dactilografia. O comando TAB determina quantos espaços devem existir em uma linha.

■ RETURN(CR)

Essa instrução movimenta para a direita 15 espaços na tela, e em seguida - imprime a palavra "início", e logo abaixo o número 15 que indica o número de espaços em branco.

■ RETURN(CR)

Responda às perguntas propostas

■ RETURN(CR)

■ RETURN(CR)

HTAB instrue o computador, para qual ponto da linha será impresso, não le vando em conta o que já foi impresso anteriormente.

■ RETURN(CR)

Responda às perguntas propostas

RETURN(CR)

Essa instrução manda o computador até a linha 10, e depois move dez espaços para a direita para imprimir X

RETURN(CR)

RETURN(CR)

Aqui terminamos a Aula 6 Parte 1. Leia o Caderno de Exercícios e pratique.
Desligue o computador. Ao ligá-lo digite o número correspondente à Aula 6-2

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1

VISUAL 1

Nesta aula você aprenderá tudo sobre variáveis, que têm como conteúdo, letras ou números.

■ RETURN(CR)

Até agora todas as variáveis que temos utilizado tiveram como conteúdo somente números.

RETURN(CR)

Como podemos recordar, variáveis que contenham números, podem ser endereçadas com um nome, utilizando qualquer letras como A,B ou C.

RETURN(CR)

Nós podemos ter variáveis que contenham letras.

■ RETURN(CR)

Estas são denominadas variáveis STRING . Estas variáveis precisam,entretanto, conter o símbolo dólar ou cifrão (\$) em seguida à uma letra a qual seria o seu endereço. Como por exemplo, A\$, B\$, ou C\$.

RETURN(CR) ■

VISUAL 3

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 4

Aqui mostraremos como colocamos letras em uma variável STRING. Observe que as informações colocadas em variáveis STRING, estão sempre entre aspas.

■ RETURN(CR)

VISUAL 5

Responda às perguntas propostas

■ RETURN(CR)

VISUAL 6

Uma variável STRING, pode ser usada como qualquer outra variavel em uma instrução PRINT,

■ RETURN(CR)

Ou, em uma instrução INPUT

■ RETURN(CR)

VISUAL 7

Responda às perguntas propostas

VISUAL 8

Quase todos os caracteres podem ser colocados em uma variável STRING, inclusive espaço. Quando esse programa é executado, o computador, imprimirá as palavras exatamente como foram fornecidas como entrada, com espaço entre elas.

■ RETURN(CR)

■ RETURN(CR)

VISUAL 9

Neste programa, é pedido ao usuário para fornecer uma variável STRING

■ RETURN(CR)

Se o usuário digitasse : amarelo

■ RETURN(CR)

O computador imprimiria amarelo exatamente como o usuário digitou-a

■ RETURN(CR)

VISUAL 10

Responda às perguntas propostas

VISUAL 11

Essas tres instruções comparam as variáveis STRING letra por letra para verificar se são iguais.

■ RETURN(CR)

Na primeira instrução, o computador imprime PRINT"correto" se elas são iguais

■ RETURN(CR)

Na 2ª instrução, o computador:"errado" se elas não forem iguais

■ RETURN(CR)

Na 3ª, se A for sim, o computador vai para a linha 100

■ RETURN(CR)

VISUAL 12

Responda às perguntas propostas

VISUAL 13

Como pôde ser observado, as duas variáveis STRING's precisam ser exatamente iguais ou o computador decidirá que elas são diferentes.

■ RETURN(CR)

VISUAL 14

■ RETURN(CR)

Este é o fim da Aula 6 Parte 2. Leia o Caderno de Exercícios e pratique.

Desligue o computador . Ao ligá-lo escolha o TESTE 6 .

.....

AULA 7 PARTE 1

VISUAL 1

Nesta aula, você vai aprender uma técnica para reduzir o número de linhas em um programa. A instrução utilizada é GOSUB

■ RETURN(CR)

Em alguns programas, você necessita usar a mesma rotina várias vezes em de terminados pontos do programa. Observe o programa da tela.

■ RETURN(CR)

Você não pode usar a instrução GOTO para cada bloco de código repetido, por que o programa executa algo diferente a cada instrução INPUT. No retorno ou seja a volta da instrução, seria para um novo local no programa de cada vez que ele fosse executado. Observe o código repetido. Aqui está !

Esses são três blocos com instruções idênticas cada um composto por três - linhas. Vamos colocá-los em um grupo e retirá-los.

■ RETURN(CR)

Teremos de renumerá-los, naturalmente

■ RETURN(CR)

Agora teremos uma rotina que satisfaz as necessidades de três partes do nos so programa. Vejamos como conseguimos acesso a essa rotina no programa e, - como será feito o acesso do programa para a rotina ! Nós temos que preencher as seções vagas do nosso programa matriz, com uma instrução que de ace sso para a utilização de nosso bloco de rotinas.

■ RETURN(CR)

Este acesso é feito através da instrução GOSUB.

Existe uma instrução GOSUB seguida do número da linha correspondente ao nū-mero de linha onde se inicia a nova subrotina. Vejamos o exemplo

■ RETURN(CR)

Primeiro a linha 10 envia o controle para a subrotina que se inicia na li-nha 100. Um PRINT é executado, INPUT é executado, e uma linha em branco é impressa. Agora teremos de voltar para onde viemos, senão o controle passa rã para a linha 120. Portanto, precisamos de instrução return

■ RETURN(CR)

A instrução return devolve o comando ao pōnto que queremos.

■ RETURN(CR)

VISUAL 2

Responda às perguntas propostas

VISUAL 3

Aqui está nosso programa novamente. Vamos seguir este fluxo.

■ RETURN(CR)

Observe que a instrução GOSUB é mais fácil de ser executada do que várias instruções GOTO, desde que a instrução GOSUB tenha no return a volta necessária ao andamento do programa .

Agora RUN para executar o programa

RETURN (CR)

A linha 10 envia o controle para a 100, uma rotina GOSUB, que imprime uma instrução e espera um INPUT, portanto, digitamos o número 1

RETURN(CR)

A execução volta para a linha 20 e imprime NAO. A linha 30 envia novamente a execução para a linha 100 e a sequência anterior se repete. Fornecemos outro INPUT.

- RETURN(CR)

De volta para a linha 40 SIM é impresso. A linha 50 envia a execução de volta para a subrotina e nos pede um outro INPUT o qual é fornecido novamente.

■ RETURN(CR)

Há um retorno para a linha 60, e imprime o valor da variável A. E a linha 65 termina a execução do programa.

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 4

Como um programador iniciante da linguagem BASIC, você deve colocar suas subrotinas no fim do programa principal, para evitar confusão na sequência de processamento. Também não se esqueça de colocar a instrução STOP entre o bloco principal do programa e suas subrotinas.

Observe o que ocorrerá se você esquecer este aviso

■ RETURN(CR)

Sem a instrução STOP, antes da linha 1000 o controle caí na linha 70 como se a subrotina fosse parte integrante do programa principal. O programa imprimiu novamente 1,2,3,4,5 e encontra uma instrução return que não deveria estar lá. Como nenhum GOSUB enviou a subrotina para lá, é impressa uma mensagem de erro. Então, não se esqueça de isolar suas subrotinas do programa principal.

■ RETURN(CR) ■ RETURN(CR) ■ RETURN(CR)

Responda às perguntas propostas

VISUAL 5 ■ RETURN(CR) ■ RETURN(CR)

Este é o fim da Aula 7 Parte 1. Leia o Caderno de Exercícios e pratique.
Desligue o computador. Ao ligá-lo digite o número correspondente à Aula 7
Parte 2.

AULA 7 PARTE 2

Você conhece várias maneiras de armazenar números em variáveis. Você pode usar a instrução LET e a instrução INPUT. Porém, se tiver uma grande quantidade de números, você necessita de uma técnica mais simples que a instrução LET.

VISUAL 1

Mostra duas instruções novas, READ e DATA. Quando o INTERPRETADOR DE BASIC encontra uma instrução READ, imediatamente ele vai para a primeira instrução DATA, e encontra o primeiro valor de DATA que neste caso é 2. As instruções DATA podem ser localizadas em qualquer parte do programa., porém é melhor colocá-las no início ou no final de um programa.

A combinação READ ... DATA funciona assim: Com cada instrução READ, o computador seleciona o primeiro valor de DATA, contido na instrução DATA e faz - com que, uma variável seja igual a esse valor. Cada valor de DATA é separado de um outro por uma vírgula. Vamos ver como isto funciona.

■ RETURN(CR)

Para cada READ a variável X, contida no LOOP, você observa um valor de DATA sendo armazenado na variável X e sendo impresso. Quando o computador procura ler mais dados (DATA) do que os existentes no programa, termina com uma mensagem de erro. Em cada leitura da variável, o valor do segmento de DATA é usado e não há mais retorno

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 2

Não há necessidade de ter toda sua DATA em uma só linha. Podem ser usadas quantas linhas forem necessárias.

■ RETURN(CR)

Os dois programas vistos na tela comportam-se de maneira igual, apesar do fato que a única instrução DATA no programa da esquerda é dividida em 4 instruções de DATA no programa da direita. O importante é lembrar-se que a DATA (dados) são lidos um por vez, por variável e em sequência. Vírgulas entre os dados contidos em uma só linha de DATA são necessárias.

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 3

■ RETURN(CR)

Podem existir casos onde você deseja reutilizar dados, DATA, que já foram usados é então que se utiliza a instrução RESTORE;(restabelecer) ;

Observe o último fragmento de DATA em linha 50, seu valor é -1.

■ RETURN(CR)

Nós adicionamos uma linha para testar se Y é igual a -1. Se for igual, o computador será dirigido ao reestabelecimento desse fragmento no início da instrução DATA novamente na linha 40.

■ RETURN(CR) ■ RETURN(CR)

Responda às perguntas propostas

VISUAL 4

Quando você utiliza mais de uma variável, em uma instrução READ, separando cada variável por uma vírgula, os números são armazenados nas variáveis em grupos.

■ RETURN(CR)

Observe aqui, cada variável X e Y são usadas em sentido alternado da linha de dados, sendo cada X e Y impresso na tela. Por sinal uma outra maneira de evitar mensagem de erro é para testar a existência do último fragmento de dados na variável posterior, e terminar o programa antes de uma outra instrução READ.

■ RETURN(CR)

Deste modo, seu último fragmento de dados, pode assumir a função de um ponteiro, para eliminar as execuções das instruções READ.

■ RETURN(CR)

VISUAL 5

Responda às perguntas propostas

VISUAL 6

Logicamente, você pode utilizar variáveis do tipo STRING (corrente de caracteres) também.

■ RETURN(CR)

Vírgulas neste caso, são utilizadas também para separar dados. Mas, quando as vírgulas são removidas ...

■ RETURN(CR)

Espaços entre dois dados são tratados como parte integrante dos dados. Aqui a linha inteira foi considerada como um só dado., porém supondo que você deseje vírgulas entre os dados, como parte integrante deles.

■ RETURN(CR)

Então você deve indicar que as vírgulas não são usadas como delimitadores - de fragmentos de dados. Isto é feito, colocando seus fragmentos de dados entre aspas. Tudo que estiver entre aspas será considerado como um único fragmento de dado.

■ RETURN(CR)

VISUAL 7

Responda às perguntas propostas

VISUAL 8

Este é o fim da Aula 7 Parte 2

RETURN(CR)

Leia o Caderno de Exercícios e pratique. Após desligue o equipamento. Depois ligue-o e digite o número correspondente ao TESTE 7.

RETURN (CR)

[illegible]

AULA 8 PARTE 1

Em aulas anteriores você aprendeu sobre o funcionamento de **LOOPS**, programas que perfazem círculos. Nesta aula você vai aprender uma técnica de **LOOP**, fazendo uso da instrução **FOR...NEXT**.

VISUAL 1

Este é um programa contador.

■ RETURN(CR)

A medida que o programa é executado, a linha 15 primeiro fixa a variável com valor zero(0).

■ RETURN(CR)

O valor de A então é impresso pela instrução da linha 20

■ RETURN(CR)

Na linha 30 é somado 1 ao valor de A.

■ RETURN(CR)

E, na linha 40 o computador verifica se A é igual a 5. Se A é igual a 5, o programa termina. No caso de A ser igual a 1 o programa continua.

■ RETURN(CR)

A linha 50 faz o retorno para a linha 20 e o mesmo processo continua até - que a variável seja igual a 5 .

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 2

Nós também podemos contar utilizando **LOOPS** criados pela instrução **FOR...NEXT**. Primeiro fixamos o valor mínimo para a variável A: Neste caso, A é igual a zero. Depois fixamos o valor máximo para a variável A. Neste caso é igual a 4

■ RETURN(CR)

Essa instrução pode substituir as linhas 15,30,40 e 50, porque tem o mesmo desempenho como todas estas instruções combinadas.

■ RETURN(CR)

Para que o computador possa executar o **LOOP**, precisamos de uma instrução adicional. Essa instrução é denominada **NEXT** e completa o **LOOP** ou círculo.

■ RETURN(CR)

Se este programa é executado agora, conseguiremos os mesmos resultados obtidos no programa anterior.

■ RETURN(CR)

■ RETURN(CR)

VISUAL 3

Todas as instruções entre as instruções **FOR** e **NEXT** são executadas cada vez que o **LOOP** for executado. Neste caso a linha 20 imprime o valor da variável A, cada vez que o programa aumenta o valor de A

■ RETURN(CR)

RETURN (CR).

Este programa tem uma instrução PRINT entre as instruções FOR e NEXT, cada vez que se completa um LOOP a instrução PRINT, imprime o novo valor de A. O ponto e vírgula na linha 20, faz com que toda a impressão continue na mesma linha.

RETURN(CR)

Responda às perguntas propostas

Na forma da linguagem BASIC, utilizada pelo seu computador, não é necessário especificar NEXT A ou NEXT X. Ao invés você pode usar simplesmente a instrução NEXT.

RETURN(CR)

Responda às perguntas propostas

Nós podemos utilizar passos de tamanhos diferentes em LOOPS do tipo FOR... NEXT. Neste programa utilizamos passos de 10 a 50. Utilizando a instrução STEP podemos fazer o computador contar em intervalos maiores que 1.

■ RETURN(CR)

RETURN(CR)

Responda às perguntas propostas

Os LOOPS formados pelas instruções FOR...NEXT, podem até contar em ordem regressiva. Este programa conta em passos de -1 (menos um), cada vez que completa um LOOP. Observe que quando você conta em sentido regressivo, o número atribuído ao STEP é precedido pelo sinal (-) menos.

RETURN(CR)

Responda às perguntas propostas

Aqui completamos Aula 8 Parte 2. Leia o Caderno de Exercícios e pratique. Depois, desligue o computador e, ao ligá-lo digite o número da Aula 8 -2.

100

Nesta aula você vai aprender a colocar um número em uma linha de variáveis.

VISUAL 1

A linha de variáveis mostrada aqui; é denominada vetor (ARRAY). As variáveis contidas em um vetor utilizam a mesma letra seguidas por um número diferente para cada variável. Os números são colocados entre Parêntesis. A variável A entre parêntesis 1, é chamada A sub 1. A variável A (2) é denominada A sub2, e assim por diante.

■ RETURN(CR)

VISUAL 2

Responda às perguntas propostas

VISUAL 3

Um vetor pode ser usado como por exemplo, no seguinte caso: Vamos dizer que você está vendendo entradas para um show. Você possui 50 cadeiras disponíveis para cada apresentação. Haverá uma apresentação por dia durante 10 dias. Podemos visualizar este exemplo formando um vetor. Aqui temos um vetor de 10 variáveis, cada uma representando 1 dia de apresentação do show. A sub 1 é o primeiro dia de cada apresentação, A sub 10 é o último dia de apresentação. Cada uma das variáveis contém 50 entradas.

■ RETURN(CR)

Cada vez que alguém adquire uma entrada para um certo dia, você subtrai uma entrada do total daquele dia.

■ RETURN(CR)

Quando uma variável A chega ao valor zero, como no 30 dia, não se pode vender mais nenhuma entrada para esse dia.

■ RETURN(CR)

Para que dia foram vendidas todas as entradas ?

Digite o nome da variável e pressione

■ RETURN(CR)

VISUAL 4

Para utilizar vetores com mais de 11 variáveis; você precisa abrir espaço na memória do computador para o vetor. Isto é feito através da instrução DIM (dimensão).

■ RETURN(CR)

Aqui está uma instrução DIM que limita um vetor denominado A, com variáveis terminando em A sub 55.

■ RETURN(CR)

VISUAL 5

VISUAL 6

RETURN(CR)

VISUAL 7

RETURN(CR)

RETURN(CR)

VISUAL 8

RETURN(CR)

VISUAL 9

■ RETURN(CR)

VISUAL 10 ■ RETURN (CR)

Desligue o computador. Ao ligá-lo digite o número correspondente ao TESTES8.

.....

VISUAL 1

Na última aula aprendemos sobre LOOPS do tipo FOR ... NEXT. Nesta aula, vamos cobrir um aspecto do LOOP FOR...NEXT, onde um LOOP se localiza dentro do outro.

Este programa é um exemplo simples de um LOOP FOR...NEXT

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 2

Como você deve lembrar qualquer número de instrução pode ser situado entre um LOOP do tipo FOR...NEXT. Nós vamos modificar este programa, para termos 5 instruções PRINT, ao invés de uma.

■ RETURN(CR)

Cada vez que o programa completa um LOOP, as instruções de 21 a 25 são executadas e estas linhas são impressas.

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 3

Ao invés de termos 5 instruções PRINT; vamos formar um novo LOOP usando FOR ...NEXT. É assim que o LOOP se apresenta:

■ RETURN(CR)

Nós agora podemos substituir as linhas 21 até 25 com estas novas linhas. E nosso programa ficará assim:

■ RETURN(CR)

Quando este programa é executado, o computador primeiro executa o LOOP externo e faz com que A seja igual a 1. Em seguida o LOOP interno é executado e, quando sua execução se completar, o computador volta ao LOOP externo e A será fixado com o valor 2.

O mesmo processo se repete até os LOOPS externos e internos serem completamente executados.

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 4

LOOPS colocados dentro de outros LOOPS são chamados de LOOPS aninhados ou compostos. Observe em que ordem os LOOPS são executados

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 5

Um dos problemas com LOOPS compostos, é a colocação da próxima instrução, neste programa a próxima instrução para o LOOP 2 está fora de lugar.

■ RETURN(CR)

Quando este programa é executado, o computador imprime o valor de Z que é 1. Logo após, ele vai para a linha 40 e envia para o próximo valor de A ao invés de Z. Após executar o LOOP A, o computador vai para a linha 50 e, imprime uma mensagem de erro.

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 6

Uma boa regra para nunca ser esquecida, é fechar o LOOP interno com uma instrução NEXT, antes de fechar o LOOP externo.

■ RETURN(CR)

Aqui fechamos o LOOP B, e depois fechamos o LOOP A com a instrução NEXT.

■ RETURN(CR)

Como você já sabe, não é preciso usar a instrução NEXT com uma variável. Na linha 50 podemos simplesmente usar a instrução NEXT ao invés de NEXT B, e na linha 60 também.

■ RETURN(CR)

VISUAL 7

É mais fácil, entretanto, marcá-los com uma variável, quando utilizarmos LOOPS compostos, especialmente quando os programas forem mais complicados como neste caso.

■ RETURN(CR)

Responda às perguntas propostas

VISUAL 8

Este é o fim da Aula 9 Parte 1. Leia o Caderno de Exercícios e pratique. Desligue o computador, ao ligá-lo, digite o número correspondente à Aula 9 -2.

.....

AULA 9 PARTE 2

Nesta aula vamos ensiná-lo a desenhar. Os desenhos são realizados utilizando os comandos específicos fornecidos por seu computador.

■ RETURN(CR)

VISUAL 1

A tela do computador é dividida em quadrados, algo como papel quadriculado.

■ RETURN(CR)

Cada um desses quadrados pode ser ligado ou desligado pelo computador. Ligando ou desligando os quadrados há criação de padrões, como você vê aqui.

Observe esses dois quadrados. O que se pode dizer sobre eles ?

■ RETURN(CR)

Responda às perguntas propostas.

VISUAL 2

Se fossemos considerar cada quadrado negro como uma posição na tela, então poderíamos dizer que este quadrado colorido ou branco está na posição 6 no sentido horizontal e 5 no sentido vertical para abaixo.

■ RETURN(CR)

VISUAL 3

Outra maneira de dizermos a mesma coisa, é dizer que cada quadrado no módulo de gráficos, representa uma certa localização na tela. Esta localização é descrita pelo endereço, que possui 2 números. O primeiro representa a posição horizontal e diz ao computador, quantos quadrados através da tela devem ser contados. O segundo, diz quantos quadrados para baixo o computador deve contar antes de ligar um quadrado. Os números são separados por uma vírgula.

RETURN(CR) ■

VISUAL 4

Responda às perguntas propostas

VISUAL 5

No seu computador, posições gráficas são numeradas de 0 a 39 em colunas horizontais,

■ RETURN(CR)

E, de 0 a 39 em colunas verticais. (

■ RETURN(CR)

VISUAL 6

Para usar estes gráficos no seu computador, você primeiro precisa usar o comando de gráficos, que é GR.

■ RETURN(CR)

O próximo passo é para especificar qual das 16 cores você vai utilizar. Aqui utilizamos a cor nº 6.

■ RETURN(CR)

VISUAL 7

Mostramos aqui as 16 cores ou seja da cor 0 até a 16. A cor 0 é o preto

■ RETURN(CR)

VISUAL 8

Responda às perguntas propostas

VISUAL 9

O 3º passo em gráficos, é para especificar as localizações nas quais serão ligadas ou desligadas os quadrados. Você se lembra que para uma localização teremos um endereço contendo 2 números. Para ligar uma localização, você usa a instrução PLOT. Veja como funciona

■ RETURN(CR)

Aqui ligamos a localização 10 na horizontal e 10 na vertical PLOT 10,10.

■ RETURN(CR)

PLOT 20,10 liga a localização 20 posições através da tela e 10 posições para baixo.

■ RETURN(CR)

PLOT 30,20 liga o quadrado 30 posições através da tela e 20 posições para baixo.

■ RETURN(CR)

VISUAL 10

Responda às perguntas propostas

VISUAL 11

Localizações a serem marcadas podem entrar através de variáveis. Cada um desses programas ligarão uma mesma posição na tela.

■ RETURN(CR)

VISUAL 12

Responda às perguntas propostas

VISUAL 13

Além de marcar localizações individuais, você também poderá desenhar linhas em seu computador. Para desenhar uma linha horizontal você usa o comando - HLIN, seguido de 3 números. A linha desenhada aqui é realizada pelo comando HLIN 10,20 at 15. Os primeiros 2 números mostram os limites mínimo e máximo da linha e o terceiro número indica em que ponto a linha estará posicionada em sentido vertical.

■ RETURN(CR)

Linhas verticais são desenhadas de modo idêntico utilizando o comando VLIN.

■ RETURN(CR)

Novamente, os 2 primeiros números indicam os pontos externos da linha, enquanto o terceiro indica em que ponto a linha está posicionada através da tela.

■ RETURN(CR)

VISUAL 14

Responda às perguntas propostas

VISUAL 15

Combinando linhas, podemos formar figuras na tela. Este programa utiliza uma série de linhas verticais, para desenhar um bloco sólido na tela.

■ RETURN(CR)

VISUAL 16

Responda às perguntas propostas

VISUAL 17

Quando qualquer um desses programas for executado, ocorrerá isto...

RETURN (CR)

VISUAL 18

Você pode usar HLTN, VLTN e PLOT para desenhar figuras na tela. Eis aqui alguns exemplos.

RETURN (CR)

Texto e gráficos não podem ser escritos na mesma linha, quando a tela estiver em modo gráfico. Quando entramos com o comando GR a tela é construída para que existam apenas 4 linhas de texto, que estão disponíveis na parte inferior da tela e, gráficos na parte superior ao texto.

RETURN (CR)

VISUAL 19

Quando terminar de estudar gráficos, você precisa utilizar o comando TEXT, para voltar ao modo normal da tela. de outro modo ocorrerá ...

RETURN (CR)

RETURN (CR)

Quando o comando TEXT é impresso para ser usado depois do modo gráfico, a tela terá a seguinte disposição.

■ RETURN (CR)

Para limpar a tela, há necessidade de usarmos o comando HOME.

■ RETURN (CR)

VISUAL 20

Responda às perguntas propostas

VISUAL 21

Este é o fim da Aula 9. Leia o Caderno de Exercícios e pratique. Depois, desligue o computador, ao ligá-lo digite o número correspondente ao TESTE 9.

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876 877 878 879 880 881 882 883 884 885 886 887 888 889 890 891 892 893 894 895 896 897 898 899 900 901 902 903 904 905 906 907 908 909 910 911 912 913 914 915 916 917 918 919 920 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 941 942 943 944 945 946 947 948 949 950 951 952 953 954 955 956 957 958 959 960 961 962 963 964 965 966 967 968 969 970 971 972 973 974 975 976 977 978 979 980 981 982 983 984 985 986 987 988 989 990 991 992 993 994 995 996 997 998 999 1000 1001 1002 1003 1004 1005 1006 1007 1008 1009 1010 1011 1012 1013 1014 1015 1016 1017 1018 1019 1020 1021 1022 1023 1024 1025 1026 1027 1028 1029 1030 1031 1032 1033 1034 1035 1036 1037 1038 1039 1040 1

Nesta aula você vai aprender como fazer seus programas tornarem-se mais eficientes.

VISUAL 1

Cada número de linha ocupa um espaço na memória. Existe uma maneira de salvar uma parte desse espaço, colocando mais de uma instrução numa mesma linha.

■ RETURN (CR)

Para formar uma linha de instruções múltiplas, apenas separe as instruções com dois pontos (:). Os dois programas são exatamente iguais. Você pode combinar várias instruções em uma mesma linha.

■ RETURN (CR)

VISUAL 2

O programa da tela mostra uma instrução GOSUB, como parte de uma linha. O retorno do GOSUB, irá levar o controle de volta para os próximos dois pontos em seguida aos do GOSUB.

Porém, quando utilizamos o comando IF - THEN você deve ter cuidado

■ RETURN (CR)

Se a condição não é cumprida, o resto da linha por inteiro será ignorado, isto é, não será executado.

RETURN (CR) ■

VISUAL 3

Como é mostrado, se o programa for executado, e você pensa que haverá um LOOP infinito por causa da instrução GOTO, você está enganado. Como todas as instruções da linha 20 não serão executadas, desde que a condição não foi satisfeita, não ocorrerá o LOOP.

■ RETURN (CR)

■ RETURN (CR)

VISUAL 4

Respondemos com um 0 para conseguirmos como resposta NÃO.

■ RETURN (CR)

Nós avisamos ao computador que nossa idade era inferior a 18 e, ele nos avisou que não podíamos votar.

■ RETURN (CR)

Quando a variável A foi testada para ver se era igual a 1, não foi igual. - Assim o resto da linha foi ignorado, e o resto da linha nunca será executado. Por outro lado, como precisamos testar o valor igual a zero, nós deveríamos ter colocado o teste em outra linha. Do modo que está agora, o controle passa direto para a linha 30, e imprime a mensagem de erro.

ente, utilizando apenas a instrução PRINT. Obviamente você pode utilizar variáveis.

■ RETURN (CR)

Você pode fazer qualquer cálculo aritmético com exponenciação. Porém quando seu trabalho envolve um série de cálculos, você precisa se lembrar do seguinte.

■ RETURN (CR)

VISUAL 3

Elevar um número a uma potência na ordem das prioridades aritméticas, ele sempre será executado antes da multiplicação, divisão, adição, subtração.

■ RETURN (CR) ■ RETURN (CR)

VISUAL 4

Quando depararmos com o comando ,

Ele é executado assim

■ RETURN (CR)

3 elevado a 3 é, calculado primeiro e depois é feita a multiplicação do que restar.

■ RETURN (CR)

VISUAL 5

Responda às perguntas propostas

VISUAL 6

Seu computador possui a capacidade de trabalhar com números extremamente grandes.

■ RETURN (CR)

O único problema é que com números muito grandes, eles ocupam muito espaço na linha e são difíceis de serem lidos. Especialmente pelo computador que não usa pontos para marcar e facilitar a leitura.

■ RETURN (CR)

Mesmo que o computador incluísse pontos neste exemplo, o tamanho do número faz com que a visualização seja difícil. Não seria mais fácil dizer que um com seis zeros em seguida é igual a um milhão ?

■ RETURN (CR)

Ou com 9 zeros em seguida é igual a 1 bilhão ? Agora, vamos elaborar uma forma de elaborar 1 com 6 zeros em seguida, utilizando expoentes.

■ RETURN (CR)

Nós estamos utilizando exponenciação com o número 1, entretanto o número de vezes que ele é multiplicado por si mesmo dá como resultado o número 1.

■ RETURN (CR)

Vamos experimentar agora com o número 10. Pois bem, agora começa a fazer sentido. 10 elevado a qualquer valor vai nos dar a grandeza do número, em ter-

mos de 1 com zeros em seguida ,

■ RETURN(CR)

10 elevado a 3 por exemplo, é o número 1 com 3 zeros em seguida ou seja 1000
Vamos supor que queremos o número 2 com alguns zeros em seguida, não vamos mexer com 10 elevado a um expoente, e sim...

■ RETURN(CR)

E assim multiplicamos por 2, e apresentada sua abreviação. Neste exemplo, podemos iniciar nossa série de zeros depois qualquer dígito que queiramos, simplesmente multiplicando nossa abreviação por aquele dígito.

Vejamos alguns exemplos :

■ RETURN(CR)

Sempre existirá 10 elevado a algum expoente,

■ RETURN(CR)

O símbolo circunflexo continua sempre o mesmo

■ RETURN(CR)

E o de multiplicação também

■ RETURN(CR)

■ RETURN(CR)

O computador elimina estes símbolos, e os substitue pela letra E que significa expoente.

■ RETURN(CR)

Você pode ler isto como 5 vezes 10, com o expoente 7 sobre o 10 sendo que, o expoente pode conter 2 dígitos, imprimindo assim, o número 7 como 07.

■ RETURN(CR)

E, vamos aprender como podemos ter expoentes negativos. O computador coloca um sinal positivo na frente do expoente.

O caracter que está agora na tela, é um número válido e conhecido como notação científica. Este é, o método pelo qual seu computador tratará números de 9 dígitos ou mais.

■ RETURN(CR)

VISUAL 7

Responda às perguntas propostas

VISUAL 8

Nem todos os números de grande porte são constituídos por zeros em seguida a um dígito.

■ RETURN(CR)

Escreva o primeiro dígito,

■ RETURN(CR)

Ponha a seguir um ponto,

■ RETURN(CR)

Situe o resto dos números que no exemplo, não são iguais a zero depois do ponto decimal.

■ RETURN (CR)

Depois adicione E + seguido pelo número de dígitos que existirem após o último dígito do número original.

■ RETURN (CR)

O expoente é igual ao número de dígitos que restam, depois do ponto decimal. Neste caso, o valor do exponencial é 8. A idéia é, a de conseguir o primeiro dígito de um número grande para ser um número inteiro, único e depois qualifica-lo. Nós podemos traduzir uma notação científica para a tradução normal, fazendo o inverso, ou seja primeiro escrevemos o número,

■ RETURN (CR)

E então escrevemos os dígitos depois do ponto decimal,

■ RETURN (CR)

E então contamos do ponto decimal para a direita o número de dígitos necessários. Enquanto preenchemos, os espaços vazios com zeros. Uma outra maneira de representar expoentes, é verificar quantas vezes o ponto decimal precisa se movimentar para a direita para conseguirmos a notação normal.

■ RETURN (CR)

■ RETURN (CR)

VISUAL 9

Responda às perguntas propostas

VISUAL 10

Vamos agora estudar o oposto, ou seja um número tão pequeno que precisa muitas posições decimais para expressar seu tamanho em notação normal, ou seja expoentes negativos.

■ RETURN (CR)

Como pode ser visto o expoente negativo mostra quantas posições para a esquerda o ponto decimal deve ser movido, para fazer um número ser uma fração de 1. O expoente zero em qualquer número sempre o fará ser igual a 1. Agora para converter esse número em notação científica, procuramos pelo primeiro dígito que não for igual a zero e o escrevemos seguido de um ponto decimal.

■ RETURN (CR)

Observe que procuramos isolar o primeiro dígito diferente de zero, então, colocamos o restante dos dígitos depois do ponto decimal.

■ RETURN (CR)

Em seguida, é contado para trás quantas vezes o ponto decimal precisa se movimentar do primeiro dígito até o início do número.

■ RETURN (CR)

Observe que o sinal em seguida à letra E foi mudado para negativo, indicando uma exponenciação negativa. Para executar o inverso ...

RETURN (CR)

Apenas escreva os dígitos...

■ RETURN (CR)

E, então movimenta-se o ponto decimal para trás, uma posição por vez contando de um até o nosso valor do expoente, preenchendo-o com zeros se for necessário. Neste exemplo, o ponto decimal foi movimentado para trás 7 vezes.

■ RETURN (CR)

VISUAL 11

Responda às perguntas propostas

VISUAL 12

Este é o fim da Aula 10 Parte 2. Leia o Caderno de Exercícios e pratique.
Desligue o computador, ao ligá-lo, digite o número correspondente ao TESTE 10.

