ROGÉRIO COSTA PEREIRA

COMPUTES, GRILO!

COMPUTADOR PARA TODAS AS IDADES

LIVROS TÉCNICOS E CIENTÍFICOS EDITORA

COMPUTES, GRILO!

COMPUTADOR PARA TODAS AS IDADES

ROGÉRIO COSTA PEREIRA

COMPUTES, GRILO!

COMPUTADOR PARA TODAS AS IDADES

RIO DE JANEIRO
SÃO PAULO

LIVROS TÉCNICOS E CIENTÍFICOS EDITORA

Copyright ©, 1983, Rogério Costa Pereira

Proibida a reprodução dos
textos originais, mesmo parcial,
e por qualquer processo, sem
autorização do Autor e da Editora.

Comissão de Computação da LTC: Prof. Donaldo Souza Dias — Prof. Ysmar Viana — Prof. Luís de Castro Martins.
Maria Alice Barroso funcionou como revisora do texto.
O *lay-out*, a arte-final da capa e as ilustrações são de Jimmy Scott.
Waldyr dos Santos Dias e Alberto Fernando de Araújo funcionaram como revisores de provas.
José Mesquita diagramou e paginou.

CIP-Brasil. Catalogação-na-fonte
Sindicato Nacional dos Editores de Livros, RJ.

Pereira, Rogério Costa.
P495c Computes, grilo! Computador para todas as idades / Rogério Costa Pereira. — Rio de Janeiro: LTC — Livros Técnicos e Científicos Editora S.A., 1983.

1. Computadores — Literatura juvenil I. Título

CDD — 621.381.95
82-0773 CDU — 681.31

ISBN: 85-216-0274-X

Direitos reservados por:
LTC — LIVROS TÉCNICOS E CIENTÍFICOS EDITORA S.A.
Av. Venezuela, 163 — 20220
Rio de Janeiro, RJ
1983
Impresso no Brasil

Prefácio

Este material foi desenvolvido visando os jovens acima de 10 anos, sem conhecimento anterior sobre computadores, e que já tenham completado ao menos a 4ª série do primeiro grau. Pode, porém, ser usado com proveito por qualquer pessoa que queira desenvolver uma boa noção do que é e para que serve um computador.

Seu objetivo principal é dar ao leitor uma base, ao mesmo tempo conceitual e prática, que estimule de forma explícita o desenvolvimento de seu "raciocínio lógico", e facilite, no futuro, sua inevitável interação com os computadores.

O uso ideal deste livro pressupõe o acesso do estudante a um computador, através de um teclado do tipo "máquina de escrever" e um vídeo. Pressupõe ainda a disponibilidade da linguagem de programação "**BASIC**".

O estudante não precisa conhecer a linguagem "**BASIC**" mas deve ter à sua disposição um manual de referência para eventuais consultas.

Da mesma forma, não é necessário que ele conheça o computador com que vai operar, mas certamente vai necessitar de orientação prévia sobre como usar o vídeo e o teclado. Dada a variedade de terminais, computadores e sistemas de programação existentes, foi impossível incluir essa orientação neste trabalho.

A linguagem "**BASIC**" foi aqui adotada por sua simplicidade e ampla disseminação. Está disponível tanto em grandes computadores, através de terminais, como nos microcomputadores que muito em breve serão comuns em nossas casas.

Diferenças pouco significativas entre comandos e instruções "**BASIC**" aqui usados e aqueles que devem ser usados em equipamentos específicos dificilmente afetarão o uso deste material.

É recomendável, ainda, que o estudante utilize o gabarito para a execução dos diagramas de blocos (régua de fluxograma), lápis ou lapiseira, um bloco de rascunho dos grandes e uma pasta onde possa manter seu material de forma organizada.

Foi seguindo essas recomendações que o Marcelo, aos treze anos, se tornou um dos mais jovens programadores do País. Além disso, com sua inteligência e esforço, desenvolveu um pequeno sistema de controle de estoque e me deu o estímulo definitivo para a divulgação deste trabalho.

O AUTOR

Sumário

INTRODUÇÃO, IX

PRIMEIRA PARTE -- PROGRAMAÇÃO (Módulos 1 a 5)

Módulo 1 — O Computador 3

Módulo 2 — Linguagem de Programação 13

Módulo 3 — Matemática no Computador 21

Módulo 4 — "Loops" 31

Módulo 5 — Variáveis 41

Revisão dos Módulos 1 a 5 49

SEGUNDA PARTE -- APLICAÇÕES (Módulos 6 a 10)

Módulo 6 — Viabilidade 57

Módulo 7 — Levantamento 63

Módulo 8 — Desenho 71

Módulo 9 — Codificação 77

Módulo 10 — Testes 81

Revisão Final 91

Glossário de Termos 97

Introdução

Este material está dividido em duas partes.

A primeira parte é indispensável ao entendimento do que é um computador e para que ele serve. É, também, indispensável ao entendimento da segunda parte.

A primeira parte tem cinco módulos com assuntos diferentes e um sexto módulo, de revisão. No módulo de revisão, você vai encontrar perguntas que vão indicar se você aprendeu bem os assuntos dos cinco primeiros módulos. Pelo número de cada pergunta você saberá em que módulo está a resposta.

A segunda parte mostra como o computador é usado na prática. Como exemplo, é mostrado o caso de uma fábrica de automóveis. Há cinco módulos nessa parte.

Já na primeira parte, a partir do módulo 2, você estará usando o computador. Portanto, antes de começar o módulo 2, peça a alguém que explique a você como operar o computador que você vai usar.

A partir daí, é por sua conta. Você será capaz de ir usando o computador à medida que for lendo os módulos e raramente vai precisar tirar dúvidas com alguém.

Não tenha medo nem pressa. Se você sentir que o assunto está ficando enrolado, volte até o ponto onde tudo estava bem e leia de novo toda a seqüência, sem pular nada.

Vá com calma, prestando bastante atenção. Se você se cansar no meio do caminho, pare e vá fazer outra coisa. Depois você volta e continua.

Vamos começar?

Primeira Parte

PROGRAMAÇÃO

(módulos 1 a 5)

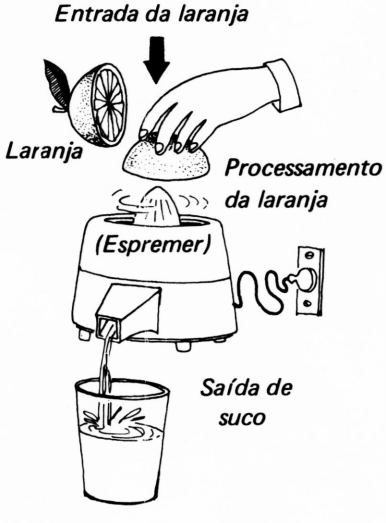
MÓDULO 1

O Computador

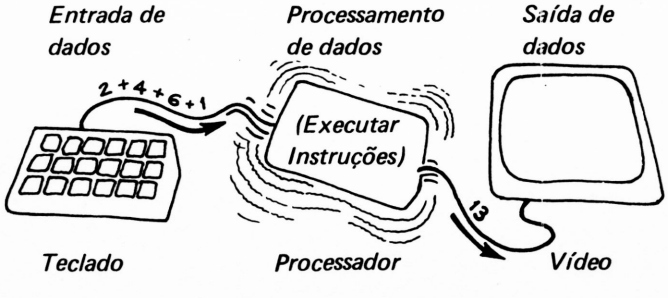
Você já viu uma máquina de moer carne?

Ela funciona assim:

E um espremedor de laranja funciona assim:

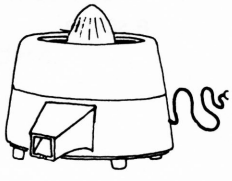


O computador funciona assim:



A máquina de moer carne
sempre processa a
carne do mesmo jeito.

O espremedor de laranja
sempre processa a
laranja do mesmo jeito.



O computador é diferente.
Ele pode processar dados de
muitas maneiras diferentes.
Depende do tipo de instruções
que a gente dê para ele.

Ele obedece as nossas
instruções e segue o
nosso raciocínio.

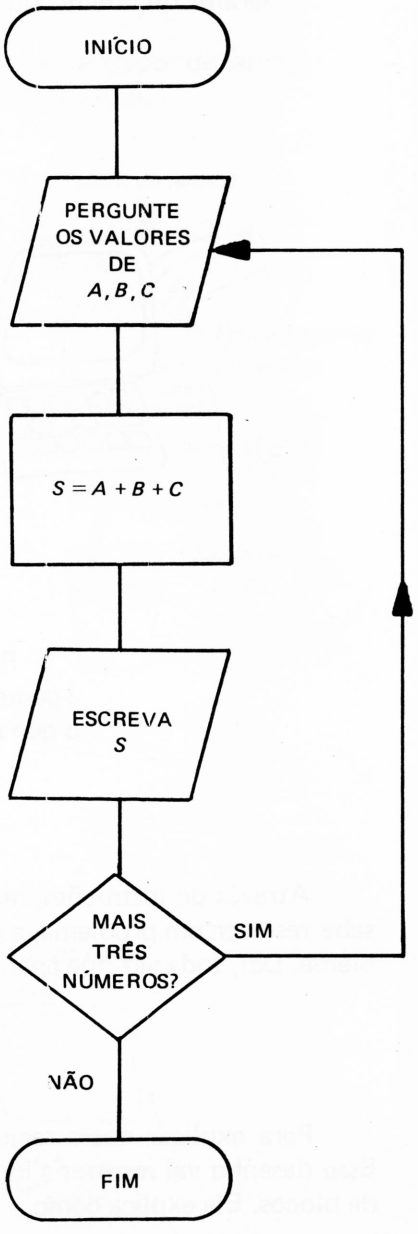
Por isso, antes de usá-lo,
a gente tem que explicar ao computador
o que a gente quer que ele faça.

Através de instruções, nós explicamos nosso raciocínio ao computador. Se a gente sabe resolver um problema, a gente pode ensinar ao computador como se resolve o problema. Daí, toda vez que aparecer esse problema, o computador resolve para a gente.

Para explicar nosso raciocínio, é preciso, antes de mais nada, fazer um desenho. Esse desenho vai mostrar a lógica do nosso raciocínio. Esse desenho se chama diagrama de blocos. Ele explica como o computador pode resolver um certo problema.

Diagrama de Blocos

Olhe bem o desenho abaixo. Ele representa a lógica que o computador deve seguir para pegar três números quaisquer, somar esses três números, e nos informar quanto dá a soma.



Quando a gente faz a lógica, ou seja, quando a gente desenha o diagrama de blocos, a gente não precisa saber quais vão ser os números que o computador vai somar. A lógica tem que servir para quaisquer números.

Como a gente não sabe os números, a gente os chama de A, B e C. Só na hora de executar o programa é que a gente vai dizer ao computador qual número é A, quanto vale B e qual é o valor de C. Daí o computador poderá fazer a soma $A + B + C$ e dizer para a gente qual é o valor de S.

Por exemplo, se $A = 1$, $B = 2$ e $C = 3$, o valor de S será 6. Se a gente quiser dar outros números para somar, a gente pode. Por exemplo, se $A = 5$, $B = 1$ e $C = 1$, o computador escreverá $S = 7$.

Programa

O computador é muito ignorante. Ele não sabe nada.

Mas ele é muito obediente. Faz tudo o que a gente manda.

A gente dá instruções ao computador e ele obedece.

A gente diz:

1Ø – LEIA OS DADOS A, B, C

2Ø – CALCULE O VALOR DE S = A + B + C

3Ø – ESCREVA O VALOR DE S

4Ø – SE HOUVER MAIS DADOS PARA LER, VOLTE PARA A INSTRUÇÃO NÚMERO 1Ø E COMECE TUDO OUTRA VEZ. SE NÃO HOUVER MAIS DADOS, PARE.

As quatro instruções acima ensinam o computador a seguir a lógica que a gente representou através do diagrama de blocos da página anterior. Elas formam um **PROGRAMA**.

Nesse programa as instruções explicam ao computador como ele deve fazer a **ENTRADA**, o **PROCESSAMENTO** e a **SAÍDA** dos dados. (Você se lembra da página 4?)

Observação – Para evitar confusão, estamos representando o algarismo zero com um corte no meio, assim: Ø. Logo, se não houver corte, não é zero, é a letra O.

Um programa é um conjunto de instruções.

Um programa diz ao computador o que ele deve fazer.

Quando o computador executa nosso programa ele está seguindo nossa lógica.

Um diagrama de blocos é a representação gráfica da lógica de um programa. Isto quer dizer que o diagrama é o desenho da lógica do programa.

Nossa lógica é mostrada às outras pessoas através do diagrama de blocos.

Nossa lógica é mostrada ao computador através do programa.

Para que serve um programa?

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Para que serve um diagrama de blocos?

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Certo ou errado:

O computador sabe muita coisa: . . . . . .

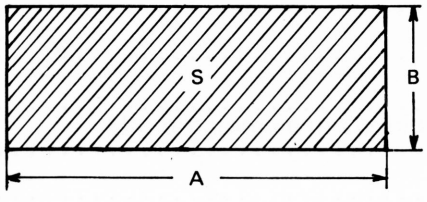
Um programa tem várias instruções: . . . . . .

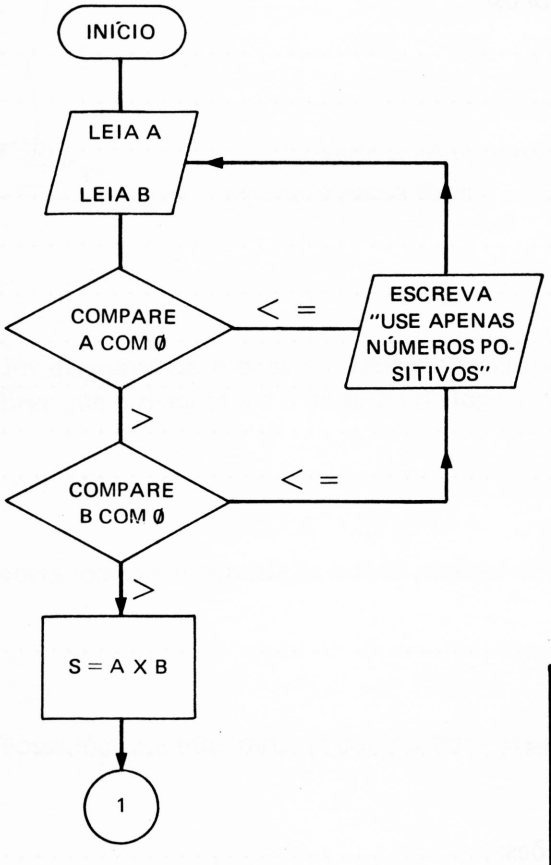
O computador obedece às instruções: . . . . . .

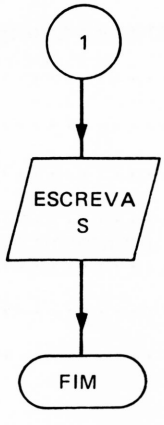
Normalmente um programa tem instruções de entrada, de processamento e de saída de dados: . . . . . .

Exemplo

A gente sabe que se um lado de um retângulo é *A* e o outro é *B*, a área do retângulo vai ser $S = A \times B$. Vamos fazer o diagrama de blocos e um programa que ensine o computador a calcular a área de um retângulo.







VOCÊ CONHECIA ESTES SINAIS?	
<	*menor que*
>	*maior que*
< =	*menor ou igual a*
> =	*maior ou igual a*

1Ø – LEIA *A*

15 – LEIA *B*

2Ø – SE *A* MENOR OU IGUAL A Ø VÁ PARA 9Ø
SE NÃO VÁ PARA A INSTRUÇÃO 3Ø

3Ø – SE *B* MENOR OU IGUAL A Ø VÁ PARA 9Ø
SE NÃO VÁ PARA 4Ø

4Ø – MULTIPLIQUE *A* POR *B* E CHAME O
RESULTADO DE *S*

5Ø – ESCREVA O VALOR DE *S*

6Ø – FIM

9Ø – ESCREVA "USE APENAS VALORES POSITIVOS"

91 – VÁ PARA 1Ø

Cada computador entende uma linguagem própria.

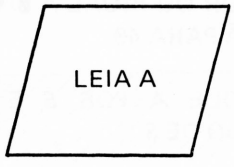
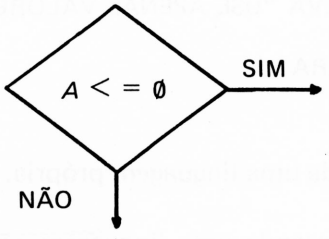
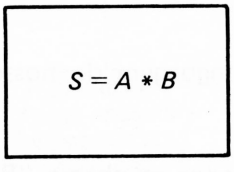
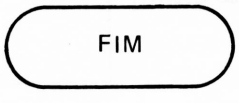
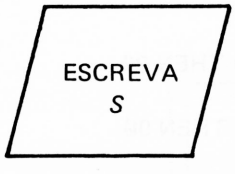
O nosso computador entende uma linguagem parecida com a que nós usamos para escrever o programa acima.

A principal diferença é que a linguagem do nosso computador usa palavras em inglês.

A linguagem do nosso computador se chama "**BASIC**". O programa acima, se for escrito na linguagem "**BASIC**", fica assim:

```
1Ø  INPUT A
15  INPUT B
2Ø  IF A < = Ø THEN 9Ø
3Ø  IF B < = Ø THEN 9Ø
4Ø  S = A * B
5Ø  PRINT S
6Ø  END
9Ø  PRINT "USE APENAS VALORES POSITIVOS"
91  GOTO 1Ø
```

Resumo

TIPO DE INSTRUÇÃO	SÍMBOLO NO DIAGRAMA DE BLOCOS	INSTRUÇÃO EM "**BASIC**"
ENTRADA DE DADOS *Exemplo*: 1Ø – LEIA *A*		1Ø INPUT *A*
PROCESSAMENTO *Exemplos*: 2Ø – VERIFIQUE O VALOR DE *A*. SE FOR IGUAL A ZERO OU MENOR, VÁ PARA A INSTRUÇÃO 9Ø. SE NÃO, VÁ PARA A PRÓXIMA INSTRUÇÃO 4Ø – MULTIPLIQUE *A* POR *B* E CHAME O RESULTADO DE *S* 6Ø FIM	  	2Ø IF *A* < = Ø THEN 9Ø 4Ø *S* = *A* * *B* 6Ø END
SAÍDA DE DADOS *Exemplo*: 5Ø – ESCREVA O VALOR DE *S*		5Ø PRINT *S*

No exemplo da página anterior, a instrução 91 – VÁ PARA 1Ø manda o computador executar a instrução número 1Ø. Em "BASIC" a gente diz 91 GOTO 1Ø. No diagrama de blocos isso é representado pelas linhas e flechas (ou setas) que mostram a seqüência que o computador deve seguir.

MÓDULO 2

Linguagem de Programação

Uma linguagem de programação é um conjunto de PALAVRAS e de SÍMBOLOS usados para se fazer os programas.

A gente usa essas palavras e símbolos para dizer ao computador o que ele deve fazer.

Exemplos de palavras usadas na linguagem "**BASIC**":

PRINT, INPUT, GOTO

Exemplos de símbolos usados na linguagem "**BASIC**":

1	2	3	4	5	6	7	8	9	Ø
A	B	?	"	$	>	<	=	+	*
U	V	X	Y	Z	(	)	E MUITOS OUTROS.		

Além das palavras e dos símbolos, uma linguagem tem REGRAS que nos dizem COMO e PARA QUE usar as palavras e os símbolos.

Quando a gente sabe programar, a gente conhece

- as palavras
- os símbolos
- as regras

da linguagem de programação.

Você se lembra de alguma regra da linguagem "BASIC"?

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Veja no programa da página 11, que palavras da linguagem "BASIC" nós já aprendemos

. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .
. . . . . .

Os símbolos são letras, algarismos e caracteres especiais. Quais foram os caracteres especiais usados no programa da página 11?

. . . . . .
. . . . . .
. . . . . .

Exercício

Escreva para o computador as instruções necessárias para que ele aprenda a calcular a área de um retângulo. Use o teclado. Vá conferindo na tela se você está escrevendo certo. Veja as instruções na página 11.

Agora execute o programa para calcular a área do retângulo (use o comando RUN). Faça A = 51 e B = 83.

Que resultado o computador achou? .....

Olhe para a tela. Imagine que você tenha trazido aqui um amigo seu que nunca viu isso antes. Você acha que ele iria entender essa tela?

.....

Para que ele visse, entendesse e achasse bacana, como deveria ser a tela?

.....

.....

Um bom programa deve fazer com que o computador e a pessoa que o está usando se comuniquem com facilidade.

Se seu programa for modificado para produzir uma tela do jeito que você disse, a pessoa e o computador vão se comunicar com facilidade?

. . . . . .

Para fazer uma tela mais bacana você tem que modificar o . . . . . .

As instruções responsáveis pela comunicação entre o computador e a pessoa são as instruções de entrada e saída.

No seu programa, quais são as instruções de entrada e saída? (Não se esqueça do número de cada instrução.)

Modifique essas instruções ou, se você achar necessário, inclua novas instruções de entrada e saída no seu programa. Quais são as instruções de entrada e saída que você vai usar?

Escreva aqui o programa modificado, inteiro.

Agora entre com ele no computador e execute-o.

Exercício

Usando a régua de fluxograma, faça o diagrama de blocos do programa, já incluindo as modificações que você fez.

MÓDULO 3

Matemática no Computador

A gente já viu que o computador é muito bobo. Ele só faz o que a gente manda.

Mas ele é muito rápido. Um computador grande é capaz de executar mais de 10 milhões de instruções em apenas 1 segundo!

Por isso, quando a gente tem cálculos muito complicados, que a gente levaria muito tempo para fazer, um computador pode ser a nossa salvação.

Além de ser muito rápido, o computador é muito preciso. Ele faz *exatamente* o que a gente manda, sem errar. Para cálculos complicados isso é muito importante, pois não se precisa perder tempo conferindo as contas.

A linguagem "**BASIC**" tem instruções para soma, subtração, multiplicação, divisão e potenciação.

Soma: se você quiser somar 6 mais 4 e fazer com que o computador escreva o resultado, basta dizer

```
PRINT 6 + 4
```

Subtração: da mesma forma, basta mandar

```
PRINT 6 – 4
```

que ele escreverá o resultado: .....

Multiplicação: nesse caso, o sinal **x** não é usado porque o computador poderia confundi-lo com a letra **x**. Por isso a gente usa o * (asterisco). A gente diz

```
PRINT 6 * 4
```

e ele nos dá o resultado: .....

Divisão: o sinal da divisão é uma barra. Assim:

```
PRINT 6/4
```

Potenciação: você sabe o que é isso? Por exemplo:

$5^2 = 5 \times 5$

$5^3 = 5 \times 5 \times 5$

$8^4 = 8 \times 8 \times 8 \times 8$

$3^2 = 3 \times 3$

$2^3 = 2 \times 2 \times 2$

É isso aí. No computador isso é representado assim:

$5 \uparrow 2$ EM VEZ DE 5^2

$5 \uparrow 3$ EM VEZ DE 5^3

$8 \uparrow 4$ EM VEZ DE 8^4

$3 \uparrow 2$ EM VEZ DE 3^2

$2 \uparrow 3$ EM VEZ DE 2^3

e a gente manda ele escrever o resultado, dizendo:

```
PRINT 5 ↑ 2
```

ele vai responder .....

```
PRINT 2 ↑ 3
```

ele vai responder .....

E se a gente quiser calcular uma expressão?

É a mesma coisa. Quer ver? Escreva:

```
PRINT 6 + 4 – 2 * 2 + 8/2 + 5 ↑ 2
```

Quanto deu? .....

Você já reparou que

```
5 + 3 * 2
```

dá resultados diferentes, dependendo da ordem em que as contas são feitas? Veja:

5 + 3 DÁ 8

8 * 2 DÁ 16

Se a gente fizer 5 + 3 * 2 ao contrário, em vez de 16 a gente vai achar 11. Veja:

2 * 3 DÁ 6

6 + 5 DÁ 11

Em Matemática, e também no computador, a ordem é sempre:

1º POTENCIAÇÃO

2º DIVISÃO E MULTIPLICAÇÃO

3º SUBTRAÇÃO E SOMA

Portanto, 5 + 4 * 3 ↑ 2 – 8/2 é calculado na seguinte ordem:

Primeiro	5 + 4 * 9 – 8/2
Depois	5 + 36 – 4
Finalmente	37, que é o resultado

Se a gente quiser que as contas sejam feitas numa ordem diferente, ou seja, se a gente não quiser

1º POTENCIAÇÃO

2º DIVISÃO E MULTIPLICAÇÃO

3º SUBTRAÇÃO E SOMA

então a gente tem que usar parênteses. Assim:

(5 + 4) * 3 ↑ 2 – (8/2)

Neste caso, o que estiver dentro dos parênteses é calculado primeiro. Assim:

9 * 3 ↑ 2 – 4

Depois de calculado o que estava nos parênteses, a ordem volta a ser

1º POTENCIAÇÃO

2º DIVISÃO E MULTIPLICAÇÃO

3º SUBTRAÇÃO E SOMA.

O computador vai achar: $9 * 9 - 4$

Depois: $81 - 4$

E finalmente 77, que é o resultado.

Quanto tinha dado sem os parênteses? .....

Quando você faz esses cálculos no computador, tudo é tão rápido que você nem percebe que ele faz as contas em etapas. Quer ver?

Mande o computador executar

PRINT 5 + 4 * 3 ↑ 2 – 8/2

Quanto deu? .....

Agora vamos usar os parênteses. Escreva:

PRINT (5 + 4) * 3 ↑ 2 – (8/2)

Quanto deu? .....

Você notou alguma diferença no tempo de cálculo? .....

E no resultado? .....

Explique por que o resultado não foi o mesmo.

.....

.....

.....

.....

.....

O sinal =, no computador, tem um significado diferente do que a gente está acostumado.

Em computador, se eu digo

A = 3 + 5

Eu estou mandando:

1º CALCULE 3 + 5

2º CHAME O RESULTADO DE A.

Eu posso então dizer coisas que em Matemática seriam absurdas, como:

A = A + 2

Para o computador, isso quer dizer:

1º PEGUE O VALOR DE A E SOME 2.

2º CHAME O RESULTADO DE A.

Qual vai ser o valor de A num programa assim:

```
20 A = 3 + 5
30 A = A + 2
40 PRINT A
```

ESCREVA AQUI:

_____

Execute no computador. Você acertou?. . . . . .

Exercício

Você vai à feira comprar bananas, laranjas, tomates e alface. Faça um programa para calcular quanto você vai gastar, sabendo os preços e as quantidades das coisas que você vai comprar.

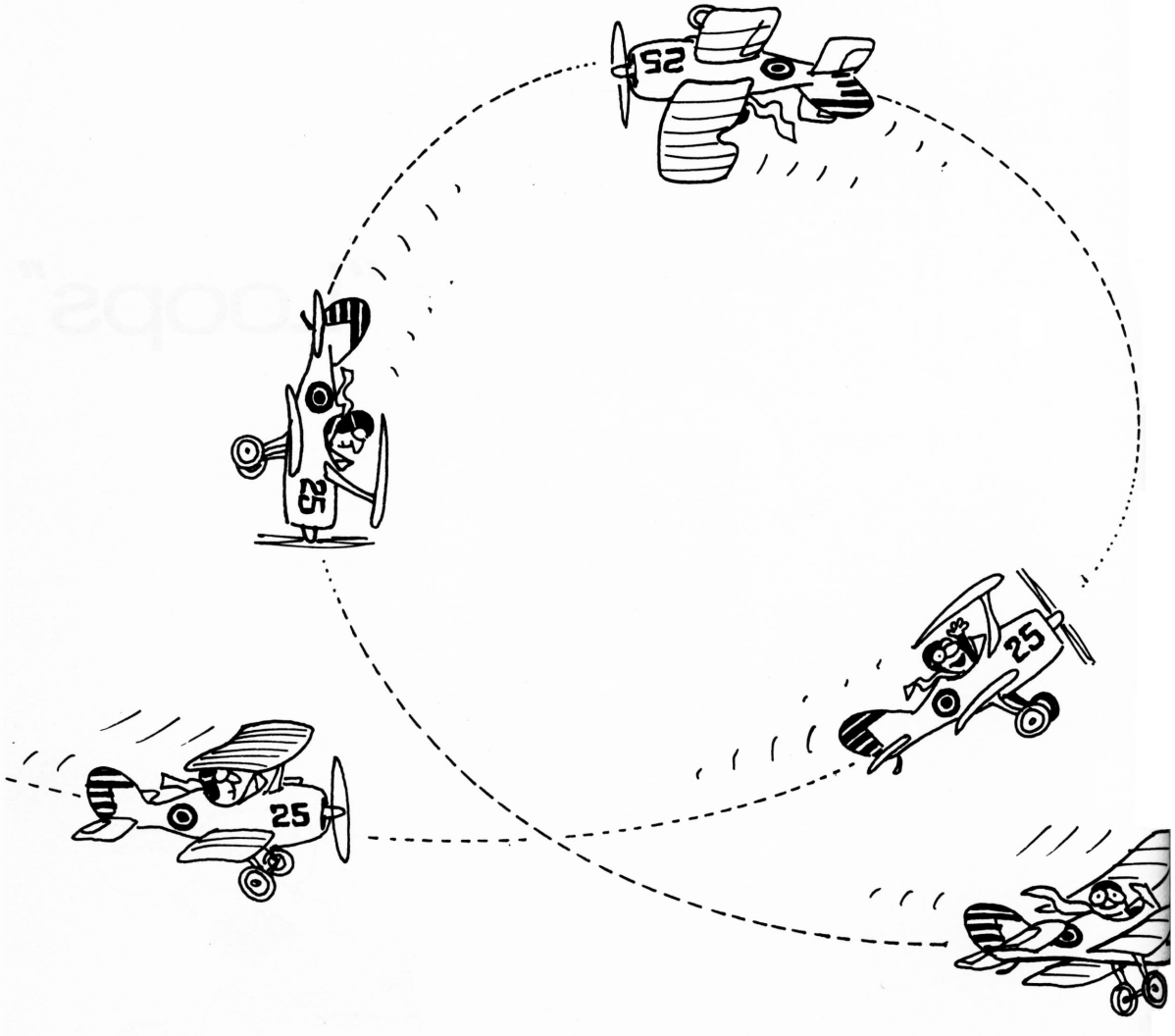
Faça primeiro o diagrama de blocos e depois as instruções. Primeiro se preocupe só com a lógica e com o processamento. Depois então capriche na comunicação (entradas e saídas). Manda brasa!

CRUZEIROS QUANTOS?
CRUZEIROS TANTOS.

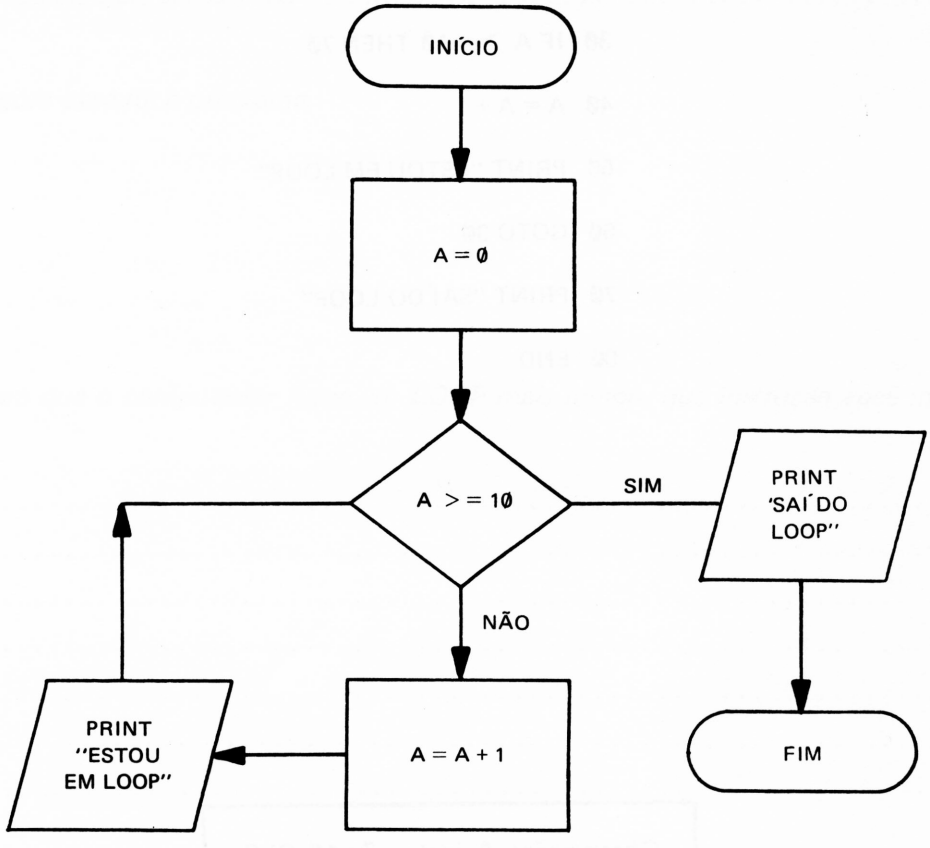
MÓDULO 4

"Loops"

Um "LOOP" (a pronúncia é lúp) é uma volta, como a que esse avião aí está fazendo:



Num programa de computador a gente também faz "LOOP". Veja um exemplo aí embaixo:



Percorra com o dedo o caminho que o computador vai seguir, do início ao fim. Você viu onde está o "LOOP"? Quando é que o computador vai sair do "LOOP"?

As instruções para esse programa poderiam ser as seguintes:

```
10 CLS
20 A = 0
30 IF A >= 10 THEN 70
40 A = A + 1
50 PRINT "ESTOU EM LOOP"
60 GOTO 30
70 PRINT "SAÍDO LOOP"
80 END
```

Observação: A instrução 10 CLS serve para "limpar" a tela. Alguns computadores usam outras instruções, em vez de CLS.

Antes de executar este programa no computador responda:

Quantas vezes ele vai escrever "estou em LOOP"? .....

Quando ele vai escrever "saí do LOOP"? .....

Agora execute o programa.

Para que o computador fique em LOOP mais tempo, que instrução você modificaria?

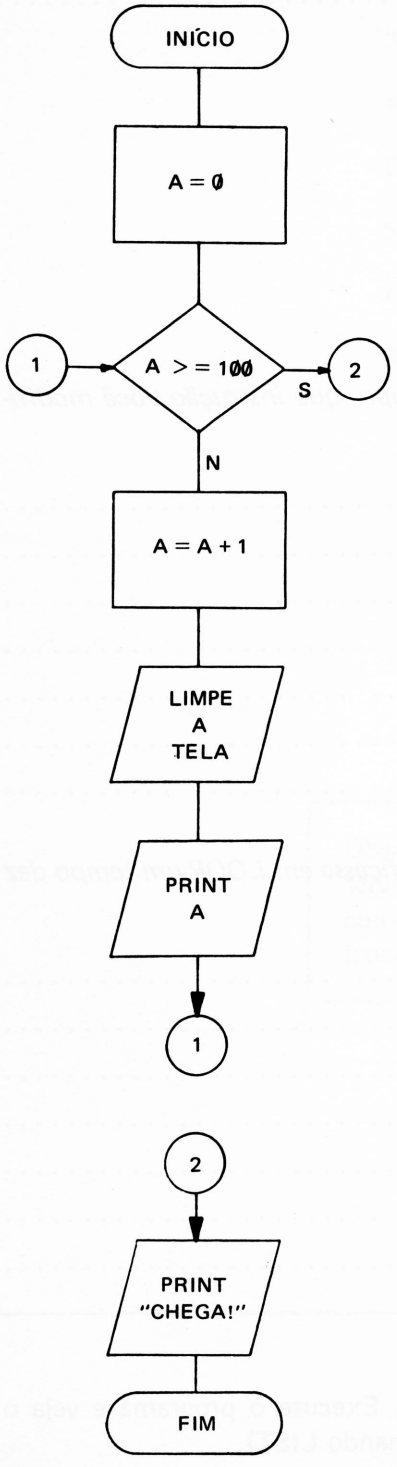
.....
.....
.....
.....
.....
.....
.....

Como seria essa instrução para que o computador ficasse em LOOP um tempo dez vezes mais longo?

.....
.....
.....
.....
.....
.....
.....

Faça essa alteração na memória do computador. Execute o programa e veja o resultado (para ver seu programa na memória, use o comando LIST).

Veja agora este outro programa. Ele é muito parecido com o que você acabou de fazer. Mas ele mostra melhor a velocidade do computador. Faça você mesmo as instruções, à direita do diagrama de blocos:



INSTRUÇÕES

1Ø . . . . . .

2Ø . . . . . .

3Ø . . . . . .

4Ø . . . . . .

5Ø . . . . . .

6Ø . . . . . .

7Ø . . . . . .

8Ø . . . . . .

Execute o programa (use o comando RUN)

Viu o computador em LOOP? .....

Marque quanto tempo leva entre o início da execução (tecla "ENTER") e o fim. Quantos segundos leva?

.....

Quantas instruções o computador executou nesse tempo? .....

(Atenção: veja na página 36. Cada vez que uma instrução for executada outra vez, você conta outra vez.)

Se você dividir o número de instruções executadas pelo tempo que o computador levou para executá-las (em segundos), você saberá quantas instruções o seu computador executou em 1 segundo.

Use o próprio computador para fazer essa divisão (veja no módulo 3 como se faz divisão no computador).

Quanto deu? ..... *instruções por segundo.*

Você ainda lembra quantas instruções por segundo um computador grande pode fazer?

.....

(Veja na página 22 do módulo 3.)

Exercício

Faça um programa para que o computador some números consecutivos, de **1** até **N**. Esse **N** é um número inteiro qualquer, que a gente só diz para o computador na hora que ele for executar o programa. **N** não pode ser menor que **1**.

Explicação:

Dado **N** = **16**, o computador vai somar

1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 1Ø +
+ 11 + 12 + 13 + 14 + 15 + 16 e imprimir
o resultado.

Dado **N** = **5**, ele soma

1 + 2 + 3 + 4 + 5 e imprime o resultado.

Se **N** = **Ø**, o computador não aceita. Se for negativo, idem.
Se **N** = **1** o computador imprime **1**.

Faça primeiro o diagrama de blocos, depois escreva as instruções.

Esse programa requer um LOOP, que você vai fazer usando "IF", "THEN" e "GOTO". Se você quiser, dê uma outra olhada no programa das páginas 33 (diagrama de blocos) e 34 (instruções). Vamos lá!

Para N = 2000 o computador executou . . . . . instruções em . . . . . segundos.

Qual foi a velocidade do computador? . . . . . instruções por segundo.

MÓDULO 5

Variáveis

Lembra do último exercício do módulo 4?

Era um programa que fazia o computador somar números consecutivos, de 1 até N.

N era um número maior ou igual a 1.

N tinha sempre o mesmo valor ou seu valor podia variar? .....

Se o valor de N pode variar, nós podemos dizer que N é UMA VARIÁVEL.

Se o valor de N fosse sempre o mesmo, N não seria uma ..... N seria UMA CONSTANTE.

Qual a diferença entre uma constante e uma variável?

— o valor de uma constante .....

— o valor de uma variável .....

Como N era um NÚMERO cujo valor podia variar, dizemos que N era uma VARIÁVEL NUMÉRICA.

Números, sejam variáveis ou constantes, podem ser somados, subtraídos, multiplicados, etc.

Por exemplo:

```
10 INPUT "QUAL O VALOR DE A"; A
20 INPUT "QUAL O VALOR DE B"; B
30 C = 2 * A + 5 * B
40 PRINT "C = "; C
50 GOTO 10
```

Execute esse programa. Depois responda:

— Quais são as variáveis que aparecem no programa? .....

E as constantes? .....

— Se eu quiser mudar o valor de uma variável basta que eu responda com o valor que eu quiser, na hora que o computador perguntar. E se eu quiser mudar o valor de uma constante? Eu tenho que mudar o .....

— Que é mais fácil, responder à pergunta do "INPUT" ou modificar o programa? .....

.....

Além das variáveis numéricas, como, A, B, C, a gente pode ter variáveis alfabéticas. Veja só (use o computador):

```
10 INPUT "RESPONDA SIM OU NÃO. VOCÊ
   ESTÁ COM SONO"; A$

20 IF A$ = "SIM" THEN 50

30 IF A$ = "NÃO" THEN 70

40 GOTO 10

50 PRINT "ENTÃO VÁ DORMIR"

60 END

70 PRINT "ENTÃO VAMOS CONTINUAR"
```

Veja a instrução 10. Nessa instrução aparece uma variável parecida com A. É a variável A$. Esse $ quer dizer que esta não é uma variável numérica. A$ é uma variável

. . . . . .

A$ é uma variável porque seu valor varia. Pode ser "sim", "não", ou qualquer outra palavra. Se for "sim", o computador desvia da instrução 20 para a instrução . . . . . *Se o valor de A$ for "não", o computador passa para a instrução 30 e desvia daí para a* . . . . . *E se A$ tiver outro valor qualquer, o que acontece?*

. . . . . .

Veja este outro exemplo:

```
10 CLS
20 INPUT "QUAL O SEU NOME"; N$
30 PRINT
40 PRINT "MUITO PRAZER"; N$
50 PRINT
```

— Para que servem as instruções 10, 30 e 50?

10: .....

30: .....

50: .....

— Existe alguma variável no programa? .....

De que tipo? .....

Qual é? .....

Não se podem fazer operações aritméticas com variáveis alfabéticas. Logo, com variáveis alfabéticas não se podem fazer SOMA, ... , ... , ... *e* .....

Mas, como você viu no programa da pág. 44, *podem-se fazer comparações. Pode-se verificar se uma variável alfabética, como A$, é ou não é igual a um certo valor, como "sim" ou "não". Isto é importante quando programamos o computador para conversar.*

Exercício

Faça um programa do tipo do que está no fim da pág. 44. O diálogo entre o computador e a pessoa é você quem tem que inventar.

10 .....

20 .....

30 .....

40 .....

50 .....

60 .....

70 .....

80 .....

Faça outro programa que pergunte o nome de quem está usando o computador. Depois, chamando a pessoa pelo nome, pergunte em que ano ela nasceu. E, finalmente, ainda chamando a pessoa pelo nome, diga quantos anos ela vai fazer.

Faça o diagrama de blocos com a lógica do seu programa e depois codifique o programa.

Revisão dos Módulos 1 a 5

Isto não é uma prova. É apenas uma oportunidade para a gente rever as coisas já vistas nos módulos 1 a 5.

Leia as questões com cuidado. Depois, se você não tiver certeza da resposta, dê uma olhada no material dos módulos 1 a 5. Só responda quando você achar que:

- entendeu a questão
- sabe a resposta

Vamos começar?

1.1) Através do raciocínio do cérebro da gente, bolamos soluções para os problemas. O computador executa essas soluções e resolve para a gente esses problemas. Ele consegue fazer isso porque nós "ensinamos" nosso raciocínio para ele. Nosso raciocínio é ensinado ao computador através de um conjunto de instruções chamado .....

1.2) Esse mesmo raciocínio pode ser expresso através de um ..... que é mais fácil para uma pessoa entender do que um conjunto de instruções.

1.3) Cada computador entende uma linguagem própria. As instruções dos nossos programas são escritas na linguagem que o nosso computador entende. Como se chama essa linguagem? .....

2.1) Num programa a gente tem, normalmente, instruções de entrada, de processamento e de saída de dados. As instruções de entrada servem para fazer o computador QUERER uma informação. Exemplo de uma instrução de ENTRADA na linguagem "BASIC":

.....

2.2) As instruções de processamento fazem com que o computador pegue as informações que ele recebeu e as transforme de acordo com nosso raciocínio, somando, subtraindo, fazendo comparações, tomando decisões; multiplicando, dividindo, fazendo potenciação, etc. Escreva uma instrução de processamento: .....

2.3) Depois que o computador recebeu e processou aquelas informações, ele chegou à solução do nosso problema. Mas a gente só fica sabendo qual é a solução, quando ele executa uma instrução de saída e nos diz o que ele tem na memória. Uma instrução de saída muito usada é .....

3.1) Uma instrução de processamento pode ser uma expressão do tipo:

$$R = 2 * 3 \uparrow 2 - 6 + 4/2$$

Os símbolos *, ↑; −, +, /, são caracteres especiais que representam as seguintes operações aritméticas:

\+ ..... − .....

* ..... / .....

↑ .....

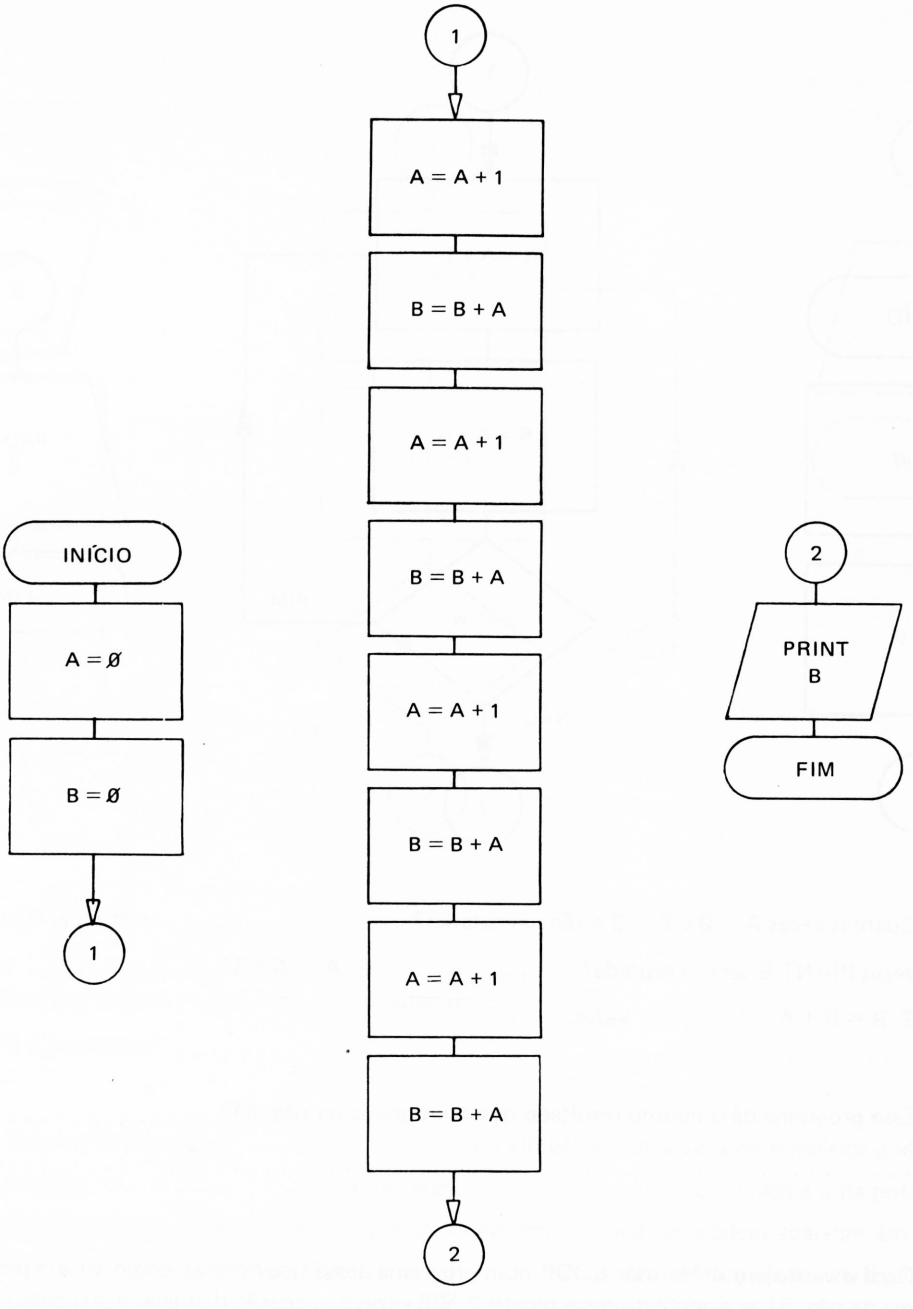
3.2) O símbolo = não é letra nem algarismo. Logo, também é um .....
.....

Em matemática ele quer dizer "IGUAL A". Mas para o computador ele quer dizer outra coisa. Como o computador entende a instrução A = 2 + 3? .....
.....

E a instrução A = A + 1? .....
.....

3.3) Qual é o resultado da expressão da questão 3.1? R = ... Se mudarmos a seqüência de cálculo, assim: R = (2 * 3) ↑ 2 − (6 + 4)/2, teremos R = ...
.....

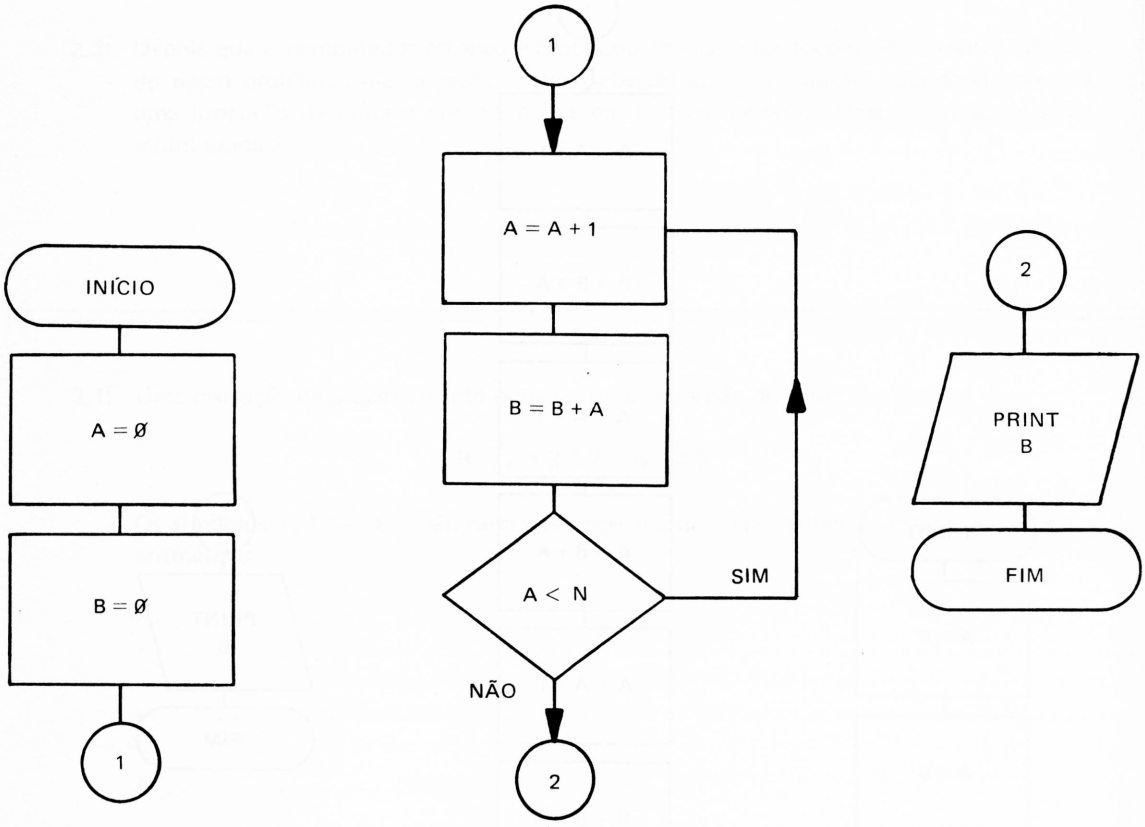
4.1) Quando a gente quer que algumas instruções, no meio de um programa, sejam executadas muitas vezes, a gente pode escrever as mesmas instruções muitas vezes. Assim:



As instruções A = Ø, B = Ø, e PRINT B serão executadas . . . . . vez cada uma.

A instrução A = A + 1 será executada . . . . . vezes. A instrução B = B + A será exe-

cutada . . . . . vezes. Em vez de escrever as mesmas instruções muitas vezes, a gente pode usar um . . . . . .



Quantas vezes A = Ø e B = Ø serão executadas? . . . . . vez cada. Quantas vezes PRINT B será executada? . . . . . E A = A + 1? . . . . . vezes. E B = B + A? . . . . . vezes.

4.2) Esse programa dá o mesmo resultado que o programa da pág. 51? . . . . . .

. . . . . .

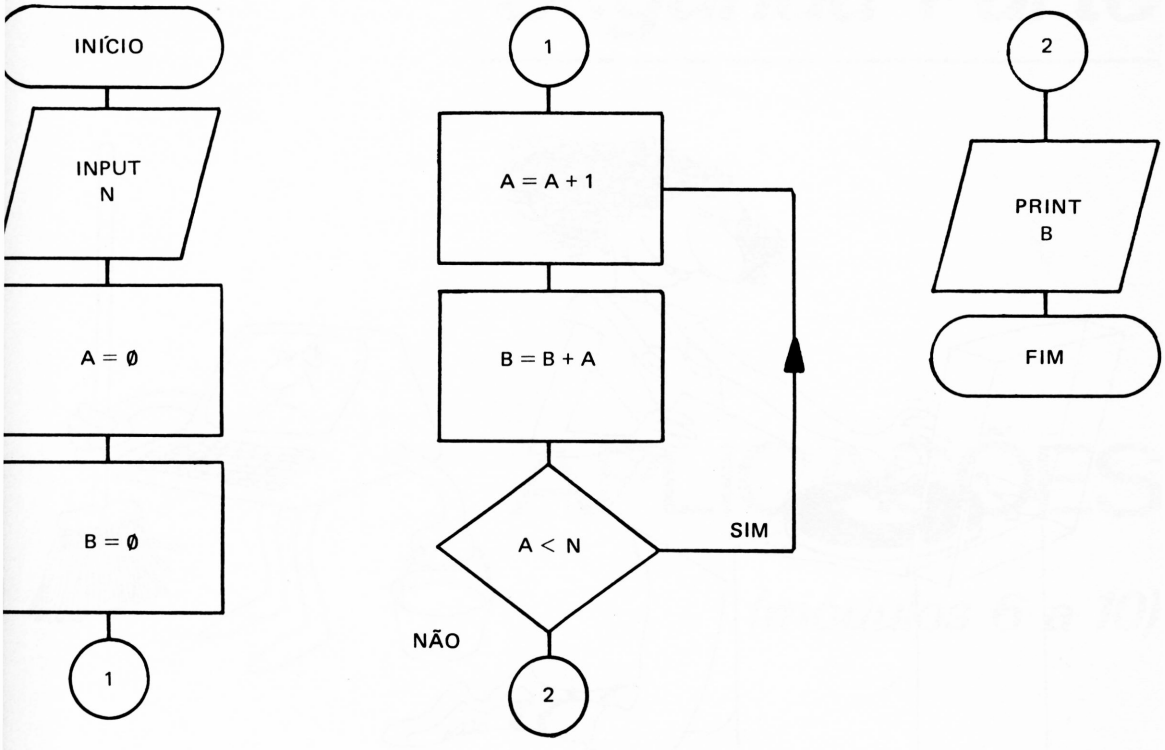
4.3) Qual a vantagem de se usar LOOP num programa desse tipo? (Pense como seria o programa da pág. 51 se a gente quisesse repetir 2.598 vezes a execução daquelas instruções.)

Qual a vantagem do LOOP? . . . . . .

. . . . . .

. . . . . .

5.1) Outra vantagem do LOOP é que a gente pode mudar à vontade o número de vezes que aquele conjunto de instruções é executado, sem ficar modificando o programa toda hora. Olha só:



Quantas vezes A = A + 1 vai ser executada? .....

N é constante? .....

5.2) Uma variável pode ser ..... ou alfabética. Uma variável numérica pode ser somada, ..., multiplicada, ..., ou elevada a uma potência qualquer. Aliás, é bom lembrar que só as variáveis numéricas podem aparecer em operações aritméticas. As variáveis ..... não podem. Mas qualquer variável, seja ..... ou ..... , pode ser comparada com uma constante ou com outra variável do mesmo tipo.

5.3) Na linguagem "BASIC" que nós usamos, como a gente indica para o computador que uma variável é alfabética? .....

Nós acabamos a primeira parte do nosso curso. Você já tem uma boa noção do que é um computador. Na segunda parte nós veremos como você pode transformar o computador num ajudante muito útil para você.

Segunda Parte

APLICAÇÕES

(módulos 6 a 10)

MÓDULO 6

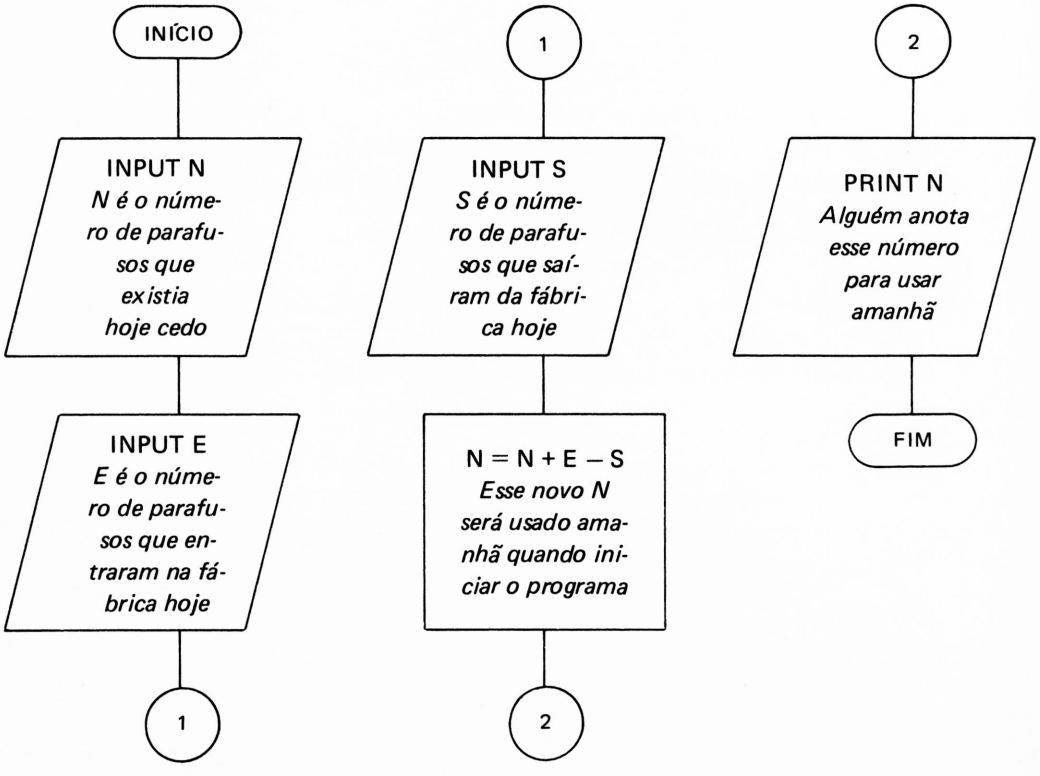
Viabilidade

Uma aplicação de computador é um uso que a gente faz do computador.

Se a gente usa o computador para controlar um foguete lançado em direção à lua, isso é uma aplicação.

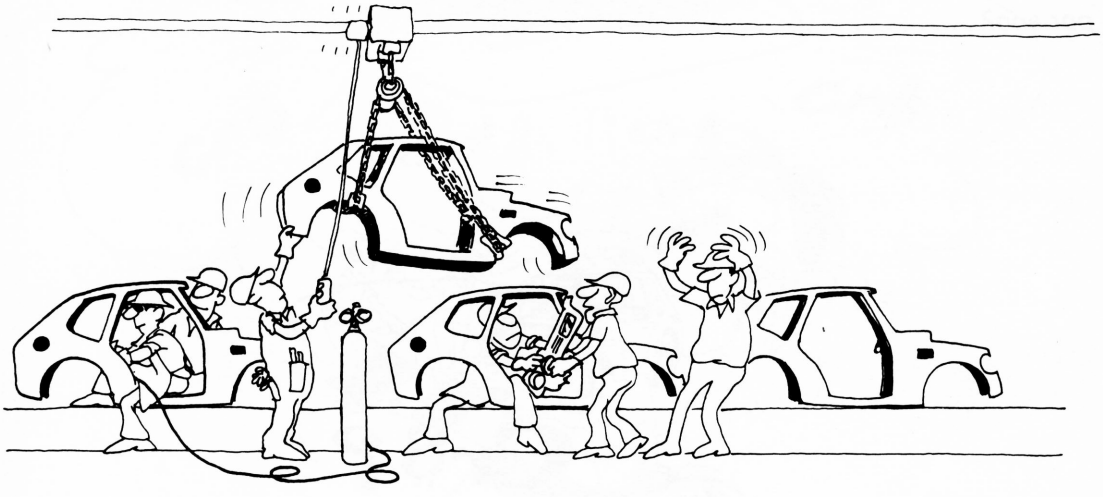
Da mesma forma, se um computador sabe quantos parafusos existem numa fábrica de automóveis, o controle do estoque de parafusos é uma aplicação.

Essa aplicação — controle do estoque de parafusos — existe de verdade. Alguém fez um programa parecido com esse aí, que faz com que o computador sempre saiba quantos parafusos existem na fábrica:



Cada automóvel que é fabricado usa um monte de parafusos que saem do depósito de parafusos quando o carro é fabricado.

O cara que toma conta dos parafusos vai anotando todos os parafusos que saem do depósito de parafusos para a linha de montagem dos carros.



Cada vez que eles vão montar um carro novo, alguém vai ao depósito e pede a quantidade de parafusos que precisa. E o cara que toma conta vai anotando. Depois ele vê quantos parafusos saíram e comunica ao computador (INPUT S).

Quando sai muito parafuso, o valor de N fica pequeno. Antes que os parafusos acabem, o cara que cuida dos parafusos deve mandar comprar mais. Quando os parafusos que ele mandou comprar chegam, ele toma nota para depois avisar ao computador (INPUT E).

Assim, graças a essa aplicação, ele não deixa nunca os parafusos acabarem e a fábrica nunca pára de fabricar carros por falta de parafusos.

Se não existisse essa aplicação e os parafusos acabassem, que aconteceria? . . . . . .

. . . . . .

Se a fábrica tinha que vender 20 carros e só fabricou 10, e por isso quem queria comprar carro comprou de outra fábrica, quantos carros ela deixou de vender? . . . . . .
Quem saiu perdendo? . . . . . .

Se cada carro vendido dá um lucro de 10 mil cruzeiros, quantos cruzeiros a fábrica perdeu por falta de uma aplicação de controle do estoque de parafusos? . . . . . .

Se isso acontece normalmente uma vez por mês, quantos cruzeiros a fábrica perde por ano? . . . . . .

A fábrica resolve alugar um computador para poder ter essa aplicação. O aluguel do computador será de 45 mil cruzeiros por mês. Muita gente na fábrica está achando caro. Você acha que eles devem alugar? . . . . . .

Por quê (Quanto eles ganhariam ou perderiam por mês se alugassem o computador e o usassem para controlar o estoque de parafusos?) . . . . . .
. . . . . .

E se em vez de custar 45 mil o computador custasse 450 mil cruzeiros, valeria a pena? . . . . . .

Isso que você fez é um estudo de viabilidade. Se a gente decide usar uma aplicação é porque a gente considerou que a aplicação é VIÁVEL, ou seja, ela vale a pena.

Quando uma empresa resolve comprar ou alugar um computador é porque ela tem planos de usar uma ou várias aplicações que vão compensar o custo do computador.

Nesse caso o computador é . . . . . .

Se o computador custasse mais do que a economia que a empresa vai fazer com as aplicações, então o computador não seria . . . . . Ele seria INVIÁVEL.

MÓDULO 7

Levantamento

Um ANALISTA DE SISTEMAS é um cara capaz de estudar um problema prático e de bolar uma solução para esse problema usando o computador.

Ele DESENVOLVE uma aplicação que resolve o problema.

Ele faz o diagrama de blocos da aplicação e dá esse diagrama a um PROGRAMADOR. O programador conhece bem uma linguagem de programação e é ele quem escreve as instruções que vão fazer com que o computador obedeça à lógica do analista de sistemas.

Esses caras, analista e programador, é que fazem com que o computador tenha uma utilidade prática. Eles são muito importantes para as empresas que usam computador e para as que fabricam e vendem computadores. Por isso eles são selecionados com muito cuidado, recebem um treinamento especial, e quando começam a trabalhar ganham um bom tutu. Vamos ver como eles trabalham.

Você é um analista de sistemas que foi contratado pela fábrica de automóveis "Corisco".

Eles estão furiosos porque mais uma vez a linha de montagem de carros parou por falta de parafusos. Você mostrou para eles que pode fazer uma aplicação que resolverá o problema. E você mostrou que a aplicação é viável.

Eles gostaram e mandaram você desenvolver a aplicação. Para isso, você tem que:

- Conhecer bem o problema
- Bolar uma solução
- Fazer o diagrama de blocos com a lógica da solução
- Codificar o programa na linguagem "BASIC" (eles não contrataram um programador. Você tem que fazer tudo sozinho)
- Testar o programa
- Mostrar o programa ao presidente da "Corisco"

Como você pode conhecer bem o problema? .....

.....

Para conhecer bem o problema você vai ver como funciona o depósito de parafusos. Você conversa com o cara que cuida dos parafusos e descobre que existem 2 tipos de parafusos: "A" e "B".

O parafuso "A" é o mais usado. Cada dia saem 10.000 parafusos do depósito para a linha de montagem.

O parafuso "B" é muito caro mas é pouco usado. Só 1.000 parafusos "B" saem do depósito de parafusos por dia.

A fábrica funciona 5 dias por semana. Você faz suas contas e descobre que eles usam . . . . . parafusos "A" por semana e . . . . . parafusos "B" por semana.

O cara que cuida do depósito disse que quando acaba um tipo de parafuso ele manda comprar mais. Acontece que a fábrica de parafusos leva uma semana para entregar os parafusos. Daí, a linha de montagem pára. Se ele pudesse adivinhar quando os parafusos vão acabar, ele mandaria comprar os parafusos uma semana antes e resolveria o problema. Mas como ele pode adivinhar? . . . . . .

. . . . . .

. . . . . .

. . . . . .

. . . . . .

Quantos parafusos "A" existem no depósito uma semana antes de acabarem? . . . .

. . . . . .

Quantos parafusos "B" existem no depósito uma semana antes de acabarem? . . . .

. . . . . .

Se ele souber quantos parafusos "A" existem no depósito hoje, ele pode saber quando eles vão acabar? . . . . . .

Por exemplo: hoje, no fim do dia, há 20.000 parafusos "A" no depósito. Quando vão acabar? . . . . . .

E se ele também souber quantos parafusos "B" existem hoje, ele também pode saber quanto tempo vai durar o estoque de parafusos "B"? . . . . . .

Por exemplo: quantos dias vai durar o estoque de 5.000 parafusos "B" que há hoje no depósito? . . . . . .

Se você encomendasse mais parafusos "A" e "B" imediatamente, você poderia evitar que a fábrica parasse? . . . . . *Por quê?* . . . . . .

Você acaba de descobrir que o cara que cuida do depósito de parafusos precisa de certas informações para poder cuidar dos parafusos.

A aplicação de computador que você vai desenvolver é um programa que vai fazer o computador dar ao cara as ..... que ele precisa para poder cuidar dos parafusos.

Você pode fazer uma lista das informações que ele gostaria de ter? Eu ajudo:

1. *Quantos parafusos "A" existem no depósito?*
2. .....
3. *Hoje é dia de comprar parafusos "A"?*
4. .....
5. .....
6. *Quantos dias vai durar o estoque de parafusos "B" se a gente não comprar mais?*

7. *Nome e endereço do fornecedor de parafuso "A".*

8. .....

9. .....

10. *Quanto custa o parafuso "B"?*

11. *Quantos parafusos "A" eu mando comprar?*

12. .....

Agora que a gente sabe quais são as necessidades de ..... do cara que cuida dos parafusos a gente pode começar a bolar como o programa vai ser.

Para simplificar um pouco o nosso trabalho, a gente pediu ao homem dos parafusos que selecionasse, junto com a gente, as informações que serão fornecidas pelo computador. Para os parafusos "A" as informações serão:

NA = número de parafusos "A" no depósito
PA = preço do parafuso "A"
DA = duração do estoque de parafusos "A"

NA, PA, DA serão três variáveis no nosso programa. Que tipo de variáveis serão essas? .....

Para os parafusos tipo "B" a gente vai precisar das mesmas informações. Logo, a gente também vai usar ..... variáveis numéricas. Como você acha que essas variáveis deveriam se chamar? ....., ..., ...

A gente acabou de fazer um LEVANTAMENTO do problema. Agora a gente sabe, em detalhes, como ajudar o cara dos parafusos.

Basta a gente fazer um programa que dê a ele as informações levantadas: .....
.....

No próximo módulo nós vamos ver como instruir o computador para que ele dê essas informações ao cara dos parafusos. Nós vamos fazer o P _ _ _ _ _ _ _ .

MÓDULO 8

Desenho

Você, analista de sistemas contratado pela "Corisco Automóveis S.A.", já mostrou para o presidente da companhia que uma aplicação de controle do estoque de parafusos é viável.

Por isso, ele decidiu que você deveria desenvolver a aplicação. Você planejou seu trabalho e escreveu num pedaço de papel tudo que você tem que fazer. Eu copiei seu plano de trabalho na pág. 65. Dê uma olhada.

Você fez um levantamento para conhecer bem o problema e concluiu que a solução é dar ao responsável pelo depósito de parafusos um conjunto de informações.

Algumas dessas informações serão fornecidas pelo computador. São as que você selecionou, junto com o homem dos parafusos. Para que o computador dê a ele essas informações você vai ter que desenvolver um programa.

Esse programa deverá ser capaz de fornecer as seguintes informações (faça uma lista, se você não lembra, olhe na pág. 68):

VARIÁVEL	SIGNIFICADO

Para que o computador possa informar o valor de cada uma dessas variáveis é preciso que alguém forneça ao computador os dados necessários e a maneira de calcular cada variável.

Os dados que o computador precisa para poder calcular o valor das variáveis são chamados DADOS DE ENTRADA. Sem os dados de entrada o computador nunca vai poder saber o valor dos DADOS DE SAÍDA.

Os dados de saída são as informações que o computador vai dar ao homem dos parafusos. São as variáveis da lista que você acabou de fazer.

A maneira do computador transformar os dados de entrada em dados de saída é o PROCESSAMENTO.

Como essas três coisas, entrada, processamento e saída, vão ser programadas, a fim de não se perder, você achou melhor fazer esta tabela:

DADOS DE SAÍDA	PROCESSAMENTO	DADOS DE ENTRADA
Variáveis Numéricas ou Alfabéticas	Como o computador consegue saber o valor de cada variável usada como dado de saída	Variáveis Numéricas ou Alfabéticas

Seu programa terá como objetivo fornecer ao homem que cuida dos parafusos os dados de saída que você escreveu na primeira coluna desta tabela.

Para isso, o computador vai pegar os dados de entrada, que você escreveu na última coluna, e processá-los do jeito que você colocou na coluna do meio.

Agora você já sabe tudo para fazer o programa. Seu programa vai ler (INPUT) os dados de entrada, processá-los de acordo com o que você escreveu na tabela da pág. 73 e, finalmente, informar (PRINT) os dados de saída ao nosso amigo dos parafusos.

Portanto, você já sabe tudo para fazer o diagrama de blocos do seu programa. O diagrama de blocos é o desenho da aplicação. Use a régua de fluxograma:

(Se necessário, continue o diagrama de blocos na pág. 75.)

(Continue aqui se necessário)

MÓDULO 9

Codificação

Você agora vai usar seu conhecimento da linguagem de programação "BASIC", para codificar o programa representado pelo diagrama de blocos que você fez no módulo 8. Pegue o diagrama e dê uma boa olhada nele. Só comece a codificar suas instruções depois de entender bem o diagrama.

Tudo bem? . . . . . Se não estiver tudo bem, leia as págs. 72 até 76. Isso só leva 10 minutos. Se você acha que não precisa, então pode começar. Minha sugestão é que você use folhas de rascunho, e depois passe a limpo aqui:

(Se necessário, continue na página seguinte).

Se necessário continue aqui.

Acabou? . . . . . Veja então no plano de trabalho da pág. 65 qual é sua próxima tarefa: . . . . . É isso que faremos no módulo 10.

MÓDULO 10

Testes

Antes de mostrar seu programa para o presidente da "Corisco" você quer ter certeza de que está tudo certo. Por isso você vai . . . . . o programa.

Entre com seu programa no computador.

Você vai executar o programa no computador e ver se tudo está funcionando bem.

Para fazer isso, você SIMULA uma situação real. você finge que é o homem dos parafusos.

Você dá dados de entrada fictícios para o computador, como se fossem de verdade, e vê se o computador se comporta como você gostaria.

E você procura simular TODAS AS SITUAÇÕES possíveis para ter certeza de que seu programa vai funcionar bem SEMPRE.

Quantas variáveis diferentes seu programa usa como dados de saída? . . . . . .
Verifique se todas estão sendo calculadas corretamente e preencha a *LISTA DE VERIFICAÇÃO* abaixo.

VARIÁVEL	*TUDO BEM?*	*VARIÁVEL*	*TUDO BEM?*

Faça as correções necessárias e repita o teste. Se estiver tudo bem, vá em frente.

Quantas telas diferentes seu programa produz? . . . . . Preencha as colunas "antes" na lista de verificação abaixo, com "sim" ou "não":

TELA Nº	*BONITA*		*FÁCIL DE ENTENDER*		*ÚTIL*	
	ANTES	*DEPOIS*	*ANTES*	*DEPOIS*	*ANTES*	*DEPOIS*

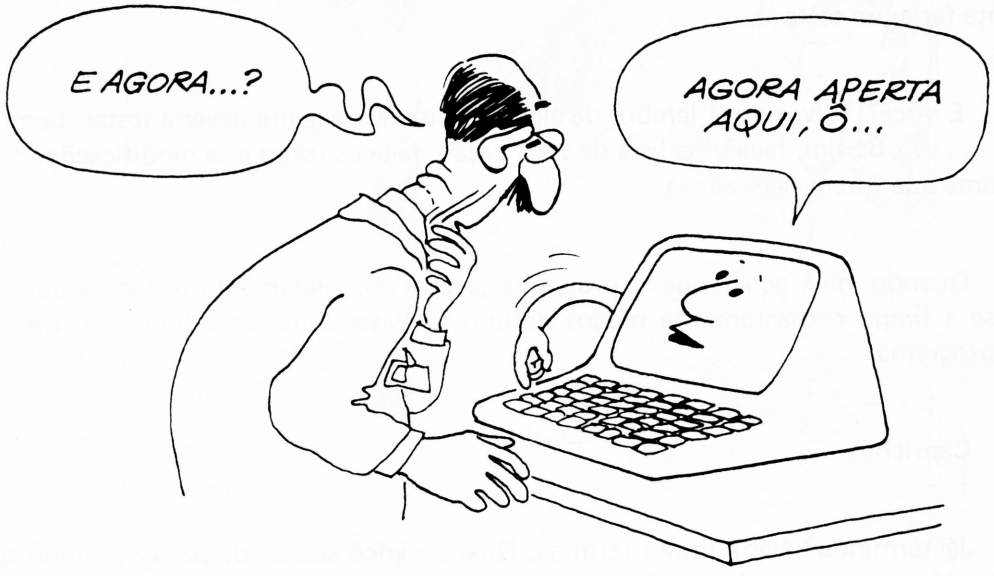
Modifique o que você achar que precisa até que todas as telas estejam como você gosta. Depois preencha as colunas "depois". Tudo bem? .....

Vamos ver agora as variáveis de entrada. Pense que o homem dos parafusos não entende nada de computador. Logo, o computador, isto é, o seu programa, tem que explicar tudo direitinho para ele. Verifique cada entrada de dados. O computador está explicando bem que dado o homem deve fornecer?

DADO DE ENTRADA (VARIÁVEL)	EXPLICAÇÃO BOA (SIM OU NÃO)

Mesmo explicando bem, o computador tem que prever que alguém pode dar para ele um dado absurdo. Nesse caso, ele deve interromper a execução normal do programa e dar uma mensagem de erro.

Se possível, ele deve orientar o nosso amigo e dar a ele uma chance. O importante mesmo é que um dado maluco não seja usado pelo computador, para evitar que ele produza uma informação .....



Experimente dar dados absurdos ao computador para ver o que acontece. Preencha a seguinte lista de verificação.

DADO DE ENTRADA (VARIÁVEL)	*VALORES ABSURDOS*	*O QUE ACONTECEU?*

Faça modificações no programa até você achar que está tudo bem.

Muito bem! Você testou entrada, processamento e saída. Isso não quer dizer que seu programa não vai errar nunca. Na vida real, pode acontecer uma situação que a gente não previu e não testou. Não me pergunte "por exemplo?". Se eu soubesse, a gente faria um teste já.

E você? Talvez você lembre de alguma coisa que a gente deveria testar. Lembrou? . . . . . Se sim, faça uma lista de verificação, faça os testes e as modificações no programa que forem necessárias.

Quando você achar que o programa já está completamente testado e aprovado, passe a limpo o diagrama de blocos definitivo. Passe também a limpo as instruções do programa.

Capriche!

Já terminou? Não? Então termine. Quando você acabar de passar a limpo o diagrama de blocos e as instruções do seu programa, você deverá se preparar para apresentar o resultado do seu trabalho ao Presidente da "Corisco".

Ele não entende muito de computador, mas é um cara muito inteligente e vai querer saber tudo sobre a aplicação que você desenvolveu.

Você quer dar a ele uma boa impressão. Para isso você vai imaginar as coisas que ele vai perguntar e já vai preparar as respostas. Assim, quando ele perguntar, você responde sem vacilar.

Será que você consegue fazer uma lista de perguntas sobre a aplicação de controle de estoques? Se você fosse o Presidente da "Corisco" o que você gostaria de saber? (Escreva a seguir as perguntas que você imaginar.)

1) .....

.....

2) .....

.....

3) .....

.....

4) .....

.....

5) .....

.....

6) .....

.....

7) .....

.....

8) .....

.....

9) .....

.....

10) .....

.....

Agora responda às perguntas da maneira mais clara e mais breve possível (o Presidente detesta gente enrolada. Vá direto ao ponto).

1) .....

.....

2) .....

.....

3) .....

.....

4) .....

.....

5) .....

.....

6) .....

.....

7) .....

.....

8) .....

9) .....

.....

10) .....

.....

Você achou difícil imaginar as perguntas do Presidente? .....

E dar as respostas? .....

Revisão Final

Se você achou difícil imaginar as perguntas que o Presidente da "Corisco" poderá fazer-lhe, esta revisão final vai quebrar o seu galho.

Você fez a revisão da primeira parte? . . . . . Esta revisão final é parecida. Não é um teste ou uma prova. Pelo contrário. É uma forma de você dar uma nova olhada em tudo o que já foi visto.

Cada pergunta tem um número. O número da pergunta indica em que módulo você poderá achar a resposta. Caso você não tenha certeza da resposta, dê uma olhada no módulo que trata do assunto.

Vamos começar?

6.1) Uma . . . . . é uma utilização que a gente faz do computador. As aplicações fazem com que os computadores sejam úteis.

6.2) Os computadores são máquinas sofisticadas. Dependendo do tamanho e da complexidade, eles podem ser muito caros. Por isso, além de serem úteis, os computadores têm que ser . . . . . Isto quer dizer que os benefícios do computador devem compensar seus custos.

6.3) Quanto maior for o número de aplicações, maior será o número de benefícios que a gente pode conseguir. Se a gente não tivesse benefícios suficientes para compensar o custo do computador, ele não valeria a pena. Ele não seria viável. Ele seria . . . . . .

7.1) Para garantir que uma aplicação vai de fato produzir benefícios, ela deve ser desenvolvida com cuidado, por pessoas preparadas para analisar um problema e bolar uma solução. Esses profissionais são os . . . . . de sistemas.

7.2) O . . . . . de . . . . . é capaz de estudar um problema prático e bolar uma solução para esse problema usando o computador. Ele . . . . . uma aplicação que resolve o . . . . . .

7.3) Uma aplicação é formada por um ou vários programas. Geralmente, o analista não escreve as instruções do programa. Ele só faz o . . . . . de . . . . . de cada programa. Ele dá esses diagramas, para um . . . . . , que é quem escreve as instruções do programa.

7.4) No seu caso, porém, você é, ao mesmo tempo, analista de sistemas e programador. Você executou um plano de trabalho com as seguintes atividades:

1 – conhecer bem o problema;

2 – bolar uma solução;

3 – fazer o diagrama de blocos com a lógica da solução;

4 – codificar o programa na linguagem "BASIC";

5 – testar o programa.

Só falta executar a última atividade:

6 – apresentar a aplicação ao Presidente.

Se você tivesse um programador trabalhando com você, que atividades você passaria para ele fazer? . . . . . .

7.5) Para conhecer bem o problema você teve que fazer um . . . . . . Quando você completou esse levantamento você ficou sabendo um monte de coisas sobre o funcionamento do depósito de parafusos. Você se lembra de quantos tipos de parafusos há no depósito? . . . . . Quais são? . . . . . Qual o mais usado? . . . . . Qual o mais caro? . . . . . .

7.6) Se o homem que cuida do depósito soubesse uma série de coisas a respeito das quantidades e da utilização dos parafusos, ele não deixaria a fábrica parar por falta de parafusos. Portanto, ele precisa que o computador lhe forneça certas . . . . . que o ajudarão a fazer seu trabalho.

7.7) Você, junto com o homem que cuida do depósito de parafusos, selecionou as informações que o computador vai fornecer a ele. Você se lembra quais são essas informações?

NA = ...
PA = ...
DA = ...
BN = ...
PB = ...
DB = ...

8.1) As informações que o computador vai fornecer ao homem que cuida do depósito dos parafusos são os dados de saída do seu programa. Para que o computador possa calcular esses dados de saída é preciso que alguém dê ao computador dados de .....

8.2) A maneira do computador transformar os dados de entrada em dados de saída é o ..... Você ensina o computador a processar os dados de entrada dando a ele um conjunto de instruções que é o seu ..... Ao executar o ....., o computador recebe dados de ..... processa esses dados e fornece ao homem dos parafusos dados de .....

8.3) A lógica do seu programa mostra como o computador recebe os dados de entrada, como ele processa esses dados, obtendo os dados de saída, e como esses dados de saída são comunicados ao homem dos parafusos. Se alguém quiser conhecer a lógica do seu programa você deverá mostrar o ..... de .....

9.1) Codificar um programa é escrever as instruções do programa. Você manda o computador receber um dado de entrada usando instruções de entrada. Dê um exemplo de uma instrução de entrada: .....

9.2) As instruções de processamento são as que fazem o computador calcular os dados de saída a partir dos dados de entrada. Dê um exemplo de uma instrução de processamento: .....

9.3) As instruções de saída fazem o computador informar à gente o resultado do processamento. Dê um exemplo de uma instrução de saída: .....

10.1) As instruções de entrada e de saída são responsáveis pela comunicação entre o homem e o computador. Como o homem dos parafusos não entende nada de computador, as instruções de entrada e de saída devem ajudá-lo a usar o computador. Como essas instruções devem ser para ajudar o homem a entender o que ele deve fazer? .....
.....
.....

10.2) Para você ter certeza da correção do seu programa, você fez alguns testes com ele. Você testou as instruções de saída para ver se as telas que elas produzem são ....., ... e .....

10.3) Você testou, também, as instruções de entrada para ver se o computador está explicando bem que dados ele quer receber. Além disso, as instruções devem evitar que, se alguém der um dado absurdo ao computador, esse dado absurdo seja processado, produzindo um dado de saída absurdo. O que o computador deve fazer se alguém lhe der um dado de entrada absurdo? . . . . . .

. . . . . .

. . . . . .

10.4) Para testar o programa você simulou a situação real em que o programa vai ser usado. O que é simular uma situação? . . . . . .

. . . . . .

. . . . . .

10.5) Para ter certeza de que você testou tudo, simulando todas as situações, você escreveu cada coisa que você queria testar. Como se chamam essas listas que você fez? . . . . . .

. . . . . .

Muito bem. Agora que você deu uma nova olhada em tudo o que foi feito, você está pronto para apresentar seu trabalho ao Presidente da "Corisco".

Glossário de Termos

Abaixo estão alguns termos usados pelas pessoas que trabalham com computador. Muitos desses termos aparecem neste livro. Aqui você encontra pequenas definições que ajudarão você a entender o significado desses termos e como eles são normalmente usados pelo pessoal.

ALGARISMOS. 1 2 3 4 5 6 7 8 9 Ø

ANALISTA DE SISTEMAS. Profissional capaz de analisar um problema e desenvolver uma solução para esse problema através do uso do computador.

APLICAÇÃO. É um uso prático que se faz do computador.

BASIC. *Ver* LINGUAGEM DE PROGRAMAÇÃO.

CARÁTER. Um símbolo, como uma letra, um algarismo ou qualquer um dos caracteres especiais (! " # $ % & ' * : = – / ? . > < , ρ ; " ↑).

CLS. Instrução BASIC. Apaga tudo o que estiver escrito na tela (limpa a tela), sem apagar a memória do computador.

CODIFICAR. Escrever as instruções de um programa.

COMANDO. Ordem isolada (diferente das instruções de um programa), dada ao computador através do teclado, para que ele faça coisas como iniciar ou interromper a execução de um programa.

COMPUTADOR. Máquina ou conjunto de máquinas para processar informações. Os computadores eletrônicos são muito rápidos e precisos. Trabalham obedecendo a instruções que damos a eles. *Ver* PROGRAMA.

CONSTANTE. Se, num programa, dizemos, por exemplo, que C = 3, isso significa que num local da memória chamado C, deve estar gravado um valor que vai ser sempre 3. C é uma constante. *Ver* VARIÁVEL.

DESENHO DE UMA APLICAÇÃO. É a concepção de uma aplicação de computador (em inglês, DESIGN).

DIAGRAMA DE BLOCOS. É a representação gráfica da lógica de um programa. *Ver* FLUXOGRAMA.

END. Instrução BASIC. Termina a execução do programa. Não é obrigatória. Mas, às vezes, é necessária.

ENTRADA. O que o computador faz para receber uma informação. A instrução de entrada que nós usamos chama-se INPUT.

ESCREVER. O computador ESCREVE uma informação na tela (ou numa impressora), quando ele executa uma instrução de saída. *Ver* SAÍDA.

ESTOQUE. A quantidade de um certo bem em poder de uma pessoa, ou de uma empresa, ou de qualquer instituição. *Exemplo*: O estoque de parafusos de uma fábrica de automóveis.

EXECUTAR UM PROGRAMA. O computador executa um programa que a gente fez. Ele obedece a cada uma das instruções do programa. *Ver* PROGRAMA e PROGRAMAR.

FLUXOGRAMA. É a representação gráfica da lógica de um programa (o mesmo que DIAGRAMA DE BLOCOS) ou, como é mais usado, a representação gráfica da ló-

gica de um conjunto de programas (Um SISTEMA).

GOTO. Instrução BASIC. Quando o computador está executando um programa e encontra essa instrução, em vez de executar a instrução seguinte, ele pula para a instrução indicada pelo GOTO.

IF . . . THEN. Instrução BASIC. Ex.: 5Ø if B = Ø then 3Ø – faz o computador executar 3Ø caso B seja Ø. Se B não for igual a Ø, ele vai executar a instrução imediatamente após a 5Ø. *Ver* IF . . . THEN . . . ELSE.

IF . . . THEN . . . ELSE. Instrução BASIC. Ex.: 5Ø if A = 3 then 9Ø else 1Ø – faz o computador desviar para 9Ø se A = 3 ou para 1Ø se A tiver outro valor (else 1Ø é opcional). *Ver* "IF . . . THEN".

INPUT. Instrução BASIC. Quando executada, faz com que o computador fique esperando que lhe seja fornecida uma informação.

INSTRUÇÃO. É uma ordem que se dá ao computador. Um conjunto de instruções forma um programa.

LER. O computador LÊ um dado quando ele obedece a uma instrução de entrada. *Ver* ENTRADA.

LEVANTAMENTO. É uma investigação que se faz para se conhecer bem um problema. É importante que se faça um levantamento do problema antes de se imaginar uma solução para ele, com ou sem computador.

LINGUAGEM DE PROGRAMAÇÃO. É um conjunto de regras de palavras e de símbolos usados para se escrever as instruções de um programa. Há várias linguagens diferentes. A que nós usamos chama-se "BASIC". *Ver* PROGRAMA.

LIST. Comando usado para se ver, na tela, as instruções do programa BASIC que está na memória do computador.

LOOP (Pronuncia-se LUP). Em um programa, é uma seqüência de instruções que se repete, formando um ciclo, uma volta, ou, como se diz em inglês, um LOOP.

MEMÓRIA. Local, dentro do computador, onde ficam gravados dados e instruções de programas. *Ver* RAM e ROM.

OPERAÇÕES ARITMÉTICAS. Soma (+), subtração (–), multiplicação (*), divisão (/) e potenciação (↑).

PRINT. Instrução BASIC. Quando executada, faz com que o computador escreva alguma coisa na tela.

PROCESSAMENTO. Tudo o que o computador faz com os dados de entrada, transformando-os através de cálculos e relacionando-os com outros dados, para obter os dados de saída.

PROGRAMA. Conjunto de instruções que damos ao computador para que ele siga nosso raciocínio e faça o que a gente quer. *Ver* LINGUAGEM DE PROGRAMAÇÃO.

PROGRAMADOR. Profissional capaz de escrever programas de computador, em uma ou mais linguagens de programação.

PROGRAMAR. Fazer um programa. Imaginar a lógica a ser seguida pelo computador e escrever instruções que o façam seguir essa lógica, quando ele executar o programa.

RAM – *RANDOM ACCESS MEMORY*. Parte da memória onde ficam gravados nossos programas e os dados que eles usam. Quando se desliga o computador, o RAM se apaga e seu conteúdo se perde.

ROM – *READ-ONLY MEMORY*. Parte da memória onde estão gravadas informações necessárias ao funcionamento do computador. Ele não grava nada no ROM, apenas lê seu conteúdo. O ROM não se apaga ao se desligar o computador.

RUN. Comando que se usa para fazer com que o computador inicie a execução de um programa.

SAÍDA. O que o computador faz para fornecer uma informação a quem o usa. PRINT é uma instrução de saída.

SIMULAÇÃO. Imitação de uma situação real.

SISTEMA. Pode significar um conjunto de máquinas ou um conjunto de programas. Na verdade, é qualquer conjunto de elementos que se AJUDAM num esforço comum para atingir um objetivo qualquer.

TELA. O conjunto de coisas escritas ou desenhadas na tela do computador ou de um terminal de computador.

TERMINAL. A um computador grande ligam-se centenas de terminais através de linhas como as dos telefones. Os terminais mais comuns têm vídeo e teclado, como os microcomputadores.

TESTE DE PROGRAMAS. Simulação das situações em que os programas vão ser usados na prática, para identificação e correção de erros, sem os riscos e prejuízos que esses erros causariam, se ocorressem na realidade.

USUÁRIO. É a pessoa que usa o computador.

VARIÁVEL. Se, em um programa, V é uma variável, V é um local da memória onde o computador vai armazenar valores que podem variar durante a execução do programa. *Ver* CONSTANTE.

VIABILIDADE. É a possibilidade de se fazer uma certa coisa. Se uma coisa é viável, ela pode ser feita.

Impresso por
Gráfica Portinho Cavalcanti Ltda.
Rua Santana, 136/138 (edifício próprio)
Tel.: 224-7732 (PABX)
Rio de Janeiro — RJ

ISBN: 85-216-0274-X

LIVROS TÉCNICOS E CIENTÍFICOS EDITORA