

Algoritmos e Programação Estruturada

Estruturas de repetição determinísticas

Você sabia que seu material didático é interativo e multimídia? Isso significa que você pode interagir com o conteúdo de diversas formas, a qualquer hora e lugar. Na versão impressa, porém, alguns conteúdos interativos ficam desabilitados. Por essa razão, fique atento: sempre que possível, opte pela versão digital. Bons estudos!

Nesta webaula vamos ver a aplicação do laço “for”, ou seja, a estrutura de repetição com variáveis de controle.

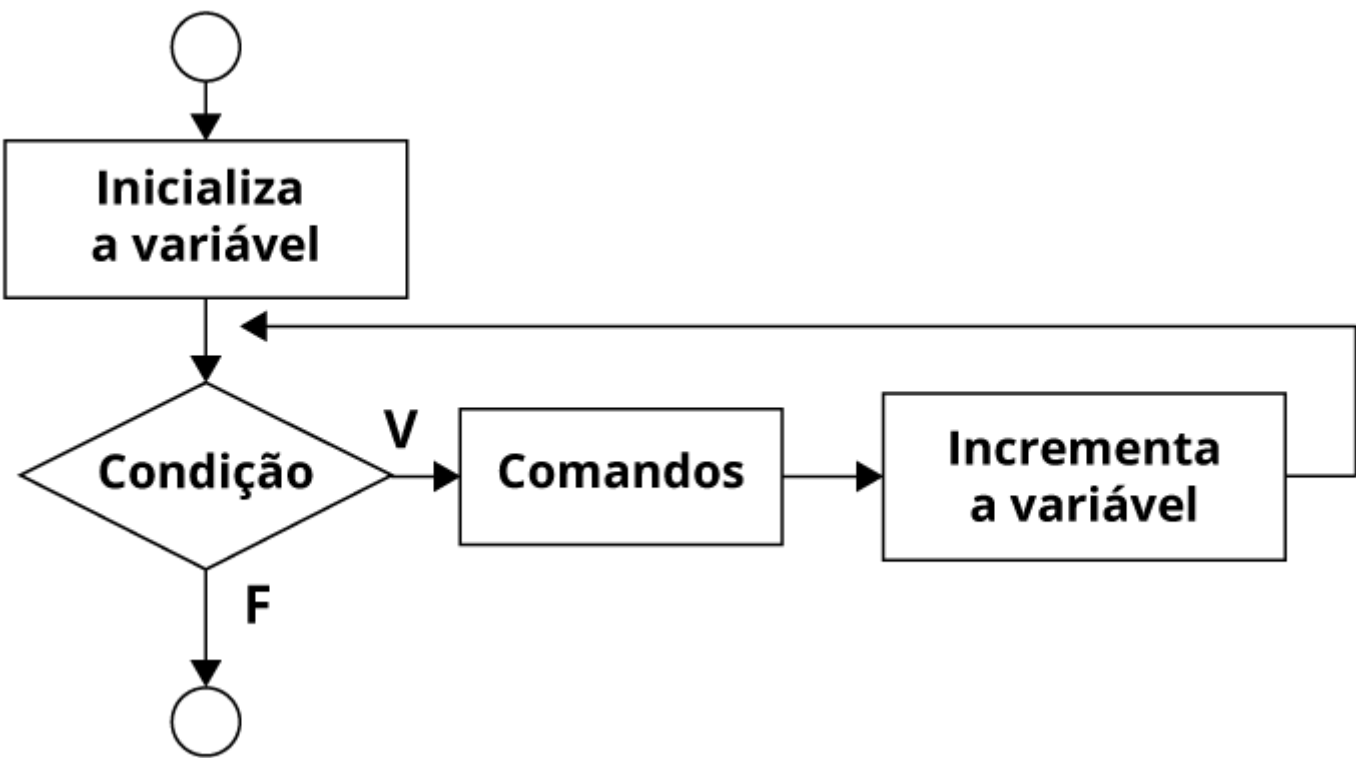
Estrutura de repetição com variáveis de controle - for

O comando iterativo “for” que em português significa “para”, é geralmente usado para repetir uma informação por um número fixo de vezes, isto é, podemos determinar quantas vezes acontecerá a repetição. Mizrahi, 2008).

Sintaxe do comando “for”:

```
for(inicialização; condição final; incremento)
{
    comandos;
}
```

Fluxograma – comando “for”



Fonte: elaborado pelo autor.

Na aplicação do comando “for” há três expressões separadas por ponto e vírgula: inicialização, condição final e incremento.

Inicialização	▼
Neste momento, coloca-se a instrução de atribuição. A inicialização é executada uma única vez antes de	

começar o laço.

Condição final

Realiza-se um teste que determina se a condição é verdadeira ou falsa; se for verdadeira, permanece no laço e, se for falsa, encerra o laço e passa para a próxima instrução.

Incremento

Parte das nossas explicações anteriores, em que é possível incrementar uma repetição de acordo com um contador específico, lembrando que o incremento é executado depois dos comandos.

Exemplos

A seguir, veremos alguns exemplos de utilização do comando `for`.

Como primeiro exemplo, iremos criar uma contagem regressiva de um número qualquer, digitado pelo usuário.

```
1  #include <stdio.h>
2  int main(void)
3  {
4      int contador;
5      printf("\nDigite um numero para contagem regressiva\n\n");
6      scanf("%d", &contador);
7      for (contador; contador >= 1; contador--)
8      {
9          printf("%d ", contador);
10     }
11     getch();
12     return(0);
13 }
```

Pode-se usar o comando “`break`” dentro de um laço “`for`” para uma determinada condição, forçando assim, o término do laço.

```
1  #include <stdio.h>
2  main()
3  {
4      int w;
5      for ( w = 1; w <= 15; w++ )
6      {
7          if ( w == 8 )
8          {
9              break;
10         }
11         printf ( “%d “, w );
12     }
13     printf( “\n \n Parar a condicao de repeticao w = %d \n”, w );
14     return 0;
15 }
```

No exemplo a seguir, temos um programa que mostra uma sequência de números, onde `x` vai de 10 a 0 e `y` vai de 0 a 10.

Representação do comando for

```
for (x = 10,y = 0; x >= 0, y <= 10 ; x--,y++)
```

Na primeira expressão “x” tem o seu valor iniciado em “10” e “y” iniciado em “0”.

Na segunda expressão o laço se repetirá enquanto n for maior ou igual a n e enquanto y for menor ou igual a 10.

Ao final da execução dos comandos do laço de repetição, x será decrementado de 1 e y será incrementado de 1.

```
1  #include <stdio.h>
2  int main()
3  {
4      int x,y;
5      for(x = 10,y = 0; x >= 0, y <= 10;
x--,y++)
6      {
7          printf("x=%2d, y=%2d\n",x,y);
8      }
9      return 0;
10 }
```

Aplicações com vetores

Vetor (*array*) é um tipo especial de variável capaz de armazenar diversos valores “ao mesmo tempo”, usando um mesmo endereço na memória.

Sintaxe: tipo variavel [n]

Na sintaxe acima [n] representa a quantidade de colunas ou linhas.

Relembre

O exemplo a seguir mescla o comando for com while. O programa encontra a primeira posição para um determinado número inserido pelo usuário.

```
1 #include <stdio.h>
2 int main()
3 {
4     int numero;
5     int i;
6     int posicao=0;
```

instrução continue `continue` dentro de um laço possibilita que a execução de comandos corrente seja terminada, passando à próxima iteração do laço.

Uma instrução continue dentro de um laço possibilita que a execução de comandos corrente seja terminada, passando à próxima iteração do laço.

No exemplo a seguir, temos um programa que percorrerá os números de 1 a 30 e neste percurso, irá testar se foi digitado algum número ímpar, caso seja ímpar o programa continua o teste até o fim do laço.

```
11 for(i=0; i<10; i++)
12 {
13     printf("\nEntre com o numero para a posicao: ");
14     scanf("%d", &numero);
15     //Preenche o vetor com numeros
16     while(vetor[posicao] != numero)
17     {
18         posicao++;
19     }
20     // Imprime vetor
21     for(i=0; i<10; i++)
22     {
23         printf("%03d ", vetor[i]);
24     }
25     // Imprime espaços até a posição do numero, e em seguida um "*" sob o numero
```

Matrizes são arranjos de duas ou mais dimensões.

Sintaxe: tipo Variável [M][N]
Onde, [M] representa a quantidade de linhas e [N] a quantidade de colunas.

O exemplo a seguir monta uma matriz 3 x 3, onde os valores são lançados de acordo com a linha e coluna.

```
1 #include <stdio.h>
2 main()
3 {
4     int i;
5     for (i=1; i <=100;i=i+1)
6     {
7         if (i==30)
8             break;
9         else
10            if (i%2==1)
11                continue;
12            else
13                printf("%2d\n",i);
14                printf("Termino do laco\n");
15    }
```

```
1 #include <stdlib.h>
2 main()
3 {
4     int linha,coluna;
5     int matriz[3][3];
6     for (linha=0; linha<3; linha++)
7     {
8         for (coluna=0; coluna<3;coluna++)
9         {
10            printf("Digitar os valores damatriz para: linha %d, coluna %d: ",linha+1,coluna+1);
11            scanf("%d", &matriz[linha][coluna]);
12        }
13    }
14    printf("Veja a sua Matriz\n");
15    for (linha=0;linha<=2;linha++)
16    {
17        for (coluna=0;coluna<3;coluna++)
18            printf("%d\t",matriz[linha][coluna]);
19        printf("\n\n");
20    }
21    system("pause");
22    return 0;
23 }
```

O vídeo referido a seguir traz uma dinâmica muito interessante na aplicação de vetores e matrizes. Realizado de “aluno para aluno”, apresenta uma revisão bem minuciosa da programação em linguagem C:

DE ALUNO PARA ALUNO. Programar em C - Revisão Vetores/Matrizes - Aula 27. 21 nov. 2012.



00:00 / 05:45



Para visualizar o vídeo, acesse seu material digital.