Algoritmos e Programação

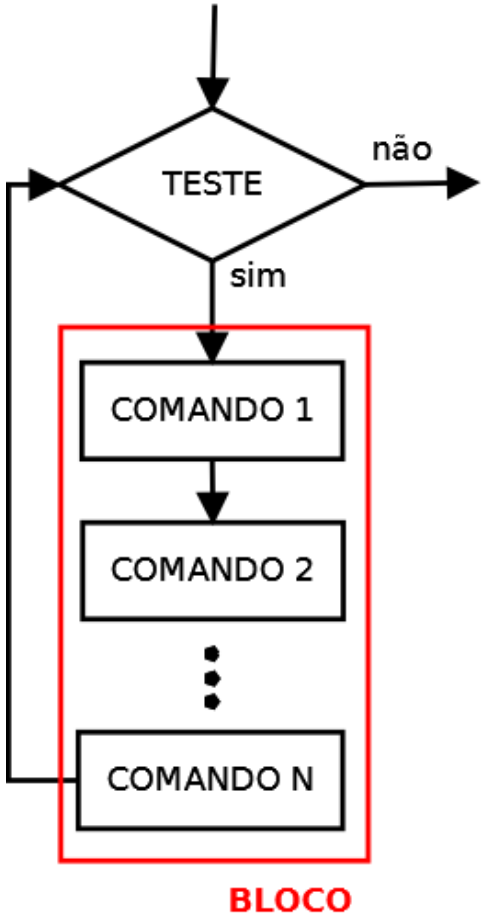
Aula 18

Estruturas de Repetição em C (*for*)

Estruturas de Repetição

Vimos que o laço de repetição ***while*** possui a seguinte sintaxe (e o equivalente fluxograma) :

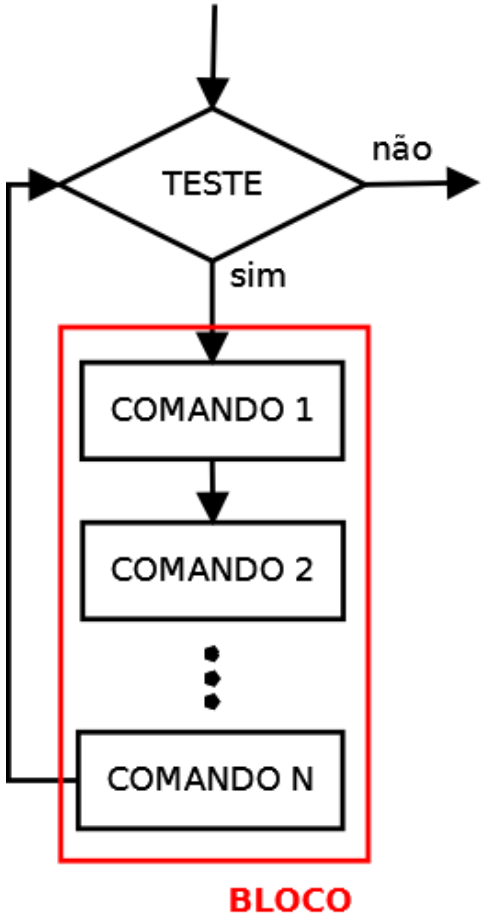
```
while (TESTE) {
COMANDO-1;
COMANDO-2;
...
COMANDO-N;
}
```



Estruturas de Repetição

Vimos que o laço de repetição ***while*** possui a seguinte sintaxe (e o equivalente fluxograma) :

```
while (TESTE) {
COMANDO-1;
COMANDO-2;
...
COMANDO-N;
}
```



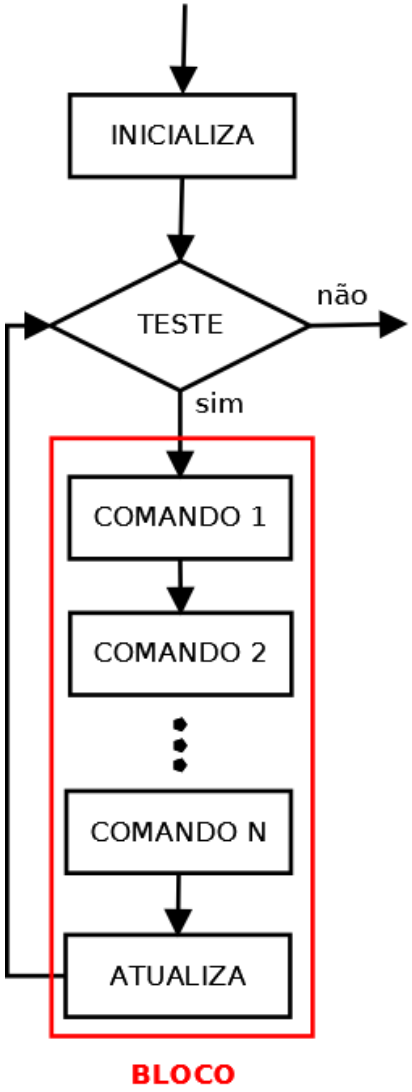
Porém, o laço **while** não possui um recurso para informar a quantidade de vezes que o laço será executado.

Estruturas de Repetição

É o programador quem deve estabelecer um método de contagem

- Para isso, utilizamos os **contadores**

```
cont = 0; //INICIALIZA
while (cont < n){//TESTA
COMANDO-1;   //comando 1
COMANDO-2;   //comando 2
...
COMANDO-N;   //comando N
cont++;      //ATUALIZA
}
```



Estruturas de Repetição

```
cont = 0; //INICIALIZA
while (cont < n){//TESTA
COMANDO-1;
COMANDO-2;
...
COMANDO-N;
cont++;      //ATUALIZA
}
```

- Esta estrutura de repetição, com **inicialização**, **teste** e **atualização**, é tão comum (e importante) que se criou um **comando específico** para **facilitar** a **escrita** deste tipo de código:

O comando **`for`**.

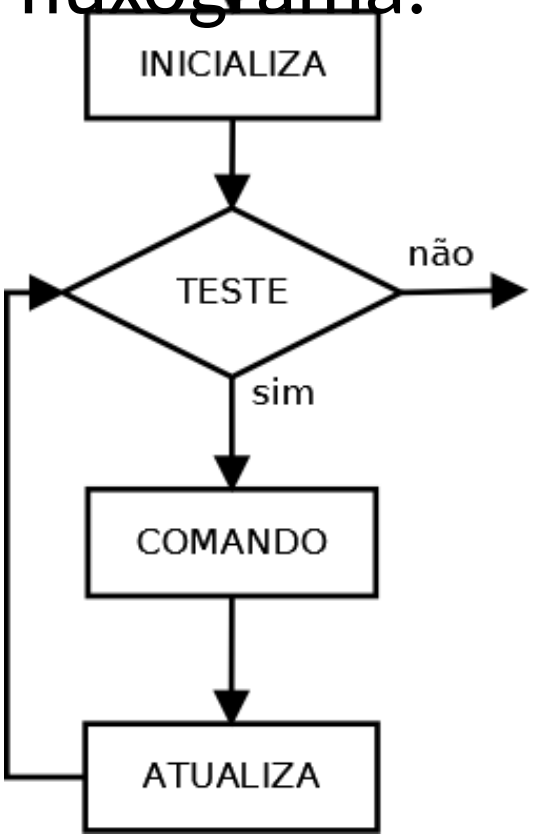
For

O comando ***for*** **facilita** a **escrita** de **laços** de repetição baseados em **contadores.**

Sintaxe:

```
for(INICIALIZA; TESTE; ATUALIZA)
COMANDO;
```

Equivale ao seguinte fluxograma:



For

O laço **`for`** é **equivalente** a um laço **`while`** de quantidade de repetições determinada **que utiliza contador.**

For

O laço **`for`** é **equivalente** a um laço **`while`** de quantidade de repetições determinada **que utiliza contador.**

Laço **`for`**:

```
for (INICIALIZA; TESTE; ATUALIZA)
COMANDO;
```

Laço **`while`** equivalente:

```
INICIALIZA;
while (TESTE) {
COMANDO;
ATUALIZA;
}
```

For

Assim como no laço ***while***, **para executarmos** um **conjunto** de **comandos** no laço ***for***, devemos **identificar** o **bloco com** o **uso** de **chaves** "{" "}".

```
for(INICIALIZA;TESTE;ATUALIZA){
COMANDO-1;
COMANDO-2;
...
COMANDO-N;
}
```

Equivalente com ***while***:

```
INICIALIZA;
while(TESTE){
COMANDO-1;
COMANDO-2;
...
COMANDO-N;
ATUALIZA;
}
```

For

A estrutura do comando ***for*** é a seguinte:

```
for (INICIALIZA; TESTE; ATUALIZA){
COMANDO-1;
COMANDO-2;
...
COMANDO-N;
}
```

Inicialização

Define o **valor inicial** do contador de iterações.

Exemplos:

i = 0

cont = 5

j = 1

For

A estrutura do comando ***for*** é a seguinte:

```
for (INICIALIZA; TESTE; ATUALIZA){
COMANDO-1;
COMANDO-2;
...
COMANDO-N;
}
```

Teste (Condição)

Define a condição de repetição. Assim como no laço while, o laço **`for`** **só é executado** se a **condição** é **verdadeira.**

Exemplos:

i <= 100

cont < 15

j == 1

For

A estrutura do comando ***for*** é a seguinte:

```
for (INICIALIZA; TESTE; ATUALIZA){
COMANDO-1;
COMANDO-2;
...
COMANDO-N;
}
```

Atualização (Incremento)

Define como o contador é incrementado.

Exemplo:

i = i + 1

cont = cont + 2

j = j – 1

i++

j--

Exemplo 1

```
#include <stdio.h>
int main() {
    int cont;
    for(cont = 1; cont <= 10; cont++)
        printf("%d\n", cont);
}
```

Inicialização

condição

Incremento

Saída:
1
2
3
4
5
6
7
8
9
10

- Contador inicia em 1
- Repete enquanto contador <= 10
- Incrementa de 1 em 1
- Este programa conta de 1 até 10 e imprime a contagem.

Exemplo 1 – Equivalente com while

```c
#include <stdio.h>
int main() {
    int cont;
    cont = 1;
    while(cont <= 10) {
        printf("%d\n", cont);
        cont++;
    }
}
```

Inicialização

condição

Incremento

Saída:
1
2
3
4
5
6
7
8
9
10

Exemplo 2

Laço ***for*** com o contador parametrizável.

```c
#include <stdio.h>
int main() {
    int i, INI, FIM, INC;
    printf("Digite INI, FIM e INC: ");
    scanf("%d %d %d", &INI, &FIM, &INC);
    for(i = INI; i <= FIM; i = i + INC){
        printf("%d\n", i);
    }
}
```

```
Exemplo 1:
INI = 5
FIM = 25
INC = 4

Saída:
5
9
13
17
21
25
-------------
Exemplo 2:
INI = 2
FIM = 8
INC = 2

Saída:
2
4
6
8
```

Exemplo 2 – Equivalente com while

Laço ***while*** com o contador parametrizável

```c
#include <stdio.h>
int main() {
    int i, INI, FIM, INC;
    printf("Digite INI, FIM e INC: ");
    scanf("%d %d %d", &INI, &FIM, &INC);
    i = INI;
    while(i <= FIM){
        printf("%d\n", i);
        i = i + INC;
    }
}
```

```
Exemplo 1:
INI = 5
FIM = 25
INC = 4

Saída:
5
9
13
17
21
25
-------------
Exemplo 2:
INI = 2
FIM = 8
INC = 2

Saída:
2
4
6
8
```

Exemplo 3

Imprime a tabuada do 1 ao 10!

```c
#include <stdio.h>
int main() {
    int i, j;
    for(i = 1; i <= 10; i++){
        for(j = 1; j <= 10; j++){
            printf("%d*%d=%d\n",i,j,i*j);
        }
        printf("\n");
    }
}
```

```
Saída:

1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
1 * 4 = 4
1 * 5 = 5
1 * 6 = 6
1 * 7 = 7
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
...


...
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100
```

Exemplo 3 – Equivalente com while

Imprime a tabuada do 1 ao 10!

```
#include <stdio.h>
int main() {
    int i, j;
    i = 1;
    while(i <= 10) {
        j = 1;
        while(j <= 10){
            printf("%d * %d =
%d\n",i,j,i*j);
            j++;
        }
        printf("\n");
        i++;
    }
}
```

```
Saída:

1 * 1 = 1
1 * 2 = 2
1 * 3 = 3
...
1 * 8 = 8
1 * 9 = 9
1 * 10 = 10

2 * 1 = 2
2 * 2 = 4
2 * 3 = 6
...


...
10 * 7 = 70
10 * 8 = 80
10 * 9 = 90
10 * 10 = 100
```

Exercício

1. Implemente um programa em C que imprime os números pares entre 2 e 26 de trás para frente, usando o comando de repetição *for*. Deve-se para isso determinar corretamente o valor de inicialização; a condição para execução; e ainda o incremento/decremento a ser realizado no *for*.

Exercício 1

```c
#include <stdio.h>
int main() {
    int N;

    for(N = 26; N >= 2; N = N - 2){
        printf("%d\n", N);
    }
}
```

Exercício

1. Implemente um programa em C que imprime os números múltiplos de 7 ou 13 entre 20 e 80. Para descobrir se um número é múltiplo use a operação que retorna o resto da divisão de dois números inteiros. Por exemplo, 8 % 2 retorna 0 ao passo que 5 % 2 retorna 1.

Exercício 2

```c
#include <stdio.h>
int main() {
    int N;
    for(N = 20; N <= 80; N++) {
        if(N % 7 == 0 || N % 13 == 0)
            printf("%d\n", N);
    }
}
```

Exercício

1. Implemente um programa em C que possui dois *for* aninhados (cada um com seu próprio contador, ex.: i e j). O programa deve imprimir o valor dos contadores i,j na forma de uma matriz. Implemente para uma matriz 4x3.

   exemplo:

```
M11  M12  M13
M21  M22  M23
M31  M32  M33
M41  M42  M43
```

Exercício 3

```c
#include <stdio.h>
int main() {
    int i, j;

    for(i = 1; i <= 4; i++){
        for(j = 1; j <= 3; j++){
            printf("M%d%d ", i, j);
        }
        printf("\n");
    }
}
```

Exercício

1. Implemente um programa em C que realiza a multiplicação de 2 números quaisquer utilizando somas sucessivas (utilize um laço *for)*.

Exercício 4

```c
#include <stdio.h>
int main() {
    int A, B, R, i;

    printf("Digite o valor de A e B: ");
    scanf("%d %d", &A, &B);
    R = 0;
    for(i = 1; i <= A ; i++){
        R = R + B;
    }
    printf("R = %d\n", R);
}
```

Exercício 4

```c
#include <stdio.h>
int main() {
    int A, B, R;

    printf("Digite o valor de A e B: ");
    scanf("%d %d", &A, &B);
    for(A, R = 0; A > 0 ; A--){
        R = R + B;
    }
    printf("R = %d\n", R);
}
```