



# ALGORITMOS E PROGRAMAÇÃO

COMANDOS DE REPETIÇÃO (LAÇO FOR) – PROF. LUCIANO ANTUNES E PROF. MATHEUS LEANDRO FERREIRA

# ESTRUTURA DE REPETIÇÃO

- Uma estrutura de repetição permite que uma sequência de comandos seja executada repetidamente, enquanto determinadas condições são satisfeitas.
- Essas condições são representadas por expressões lógicas (como, por exemplo,  $A > B$ ;  $C == 3$ ;  $\text{Letra} == \text{'a'}$ )
  - Repetição com teste no início;
  - Repetição com teste no final;
  - Repetição Contada;

## ESTRUTURA DE REPETIÇÃO

O real poder dos computadores está na sua habilidade para repetir uma operação ou uma série de operações muitas vezes.

Esta repetição, chamada laços (loop) é um dos conceitos básicos da programação estruturada;

Geralmente uma estrutura de repetição é definida como “um conjunto de comandos de um algoritmo que pode ser repetida quando subordinado a uma condição”

# COMANDO FOR

- O loop ou laço FOR é usado para repetir um comando, ou bloco de comandos, diversas vezes;
- Sua forma geral é:

```
for(inicialização; condição; incremento){  
    sequência de comandos;  
}
```

# COMANDO FOR

- **Inicialização:** iniciar variáveis (contador);
- **Condição:** avalia a condição. Se verdadeiro, executa comandos do bloco, senão encerra o laço;
- **Incremento:** ao término do bloco de comandos, incrementa o valor do contador;
  - Repete o processo até que a condição seja falsa.

```
for(inicialização; condição; incremento){  
    sequência de comandos;  
}
```

# COMANDO FOR

- Em geral, utilizamos o comando FOR quando precisamos ir de um valor inicial até um valor final;
- Para tal, utilizamos uma variável para realizar a contagem
  - **Exemplo:** `int i;`
- Nas etapas do comando FOR:
  - **Inicialização:** atribuímos o valor inicial a variável;
  - **Condição:** especifica a condição para continuar no loop
    - **Exemplo:** seu valor final
  - **Incremento:** atualiza o valor da variável usada na contagem

- Exemplo: imprime os valores de 1 até 10

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int i;
    for (i = 1; i <= 10; i++) {
        printf("%d\n", i);
    }
    system("pause");
    return 0;
}
```

The diagram illustrates the three components of a C for loop using orange boxes and arrows:

- Inicialização** (Initialization): Points to the `i = 1` part of the `for` loop.
- Condição** (Condition): Points to the `i <= 10` part of the `for` loop.
- Incremento** (Increment): Points to the `i++` part of the `for` loop.

# COMANDO FOR

# COMANDO FOR

- Enquanto o comando WHILE (que veremos nas próximas aulas) repete uma sequência de comandos até a condição ser verdadeira, o comando FOR repete uma sequência de comandos “N” vezes.

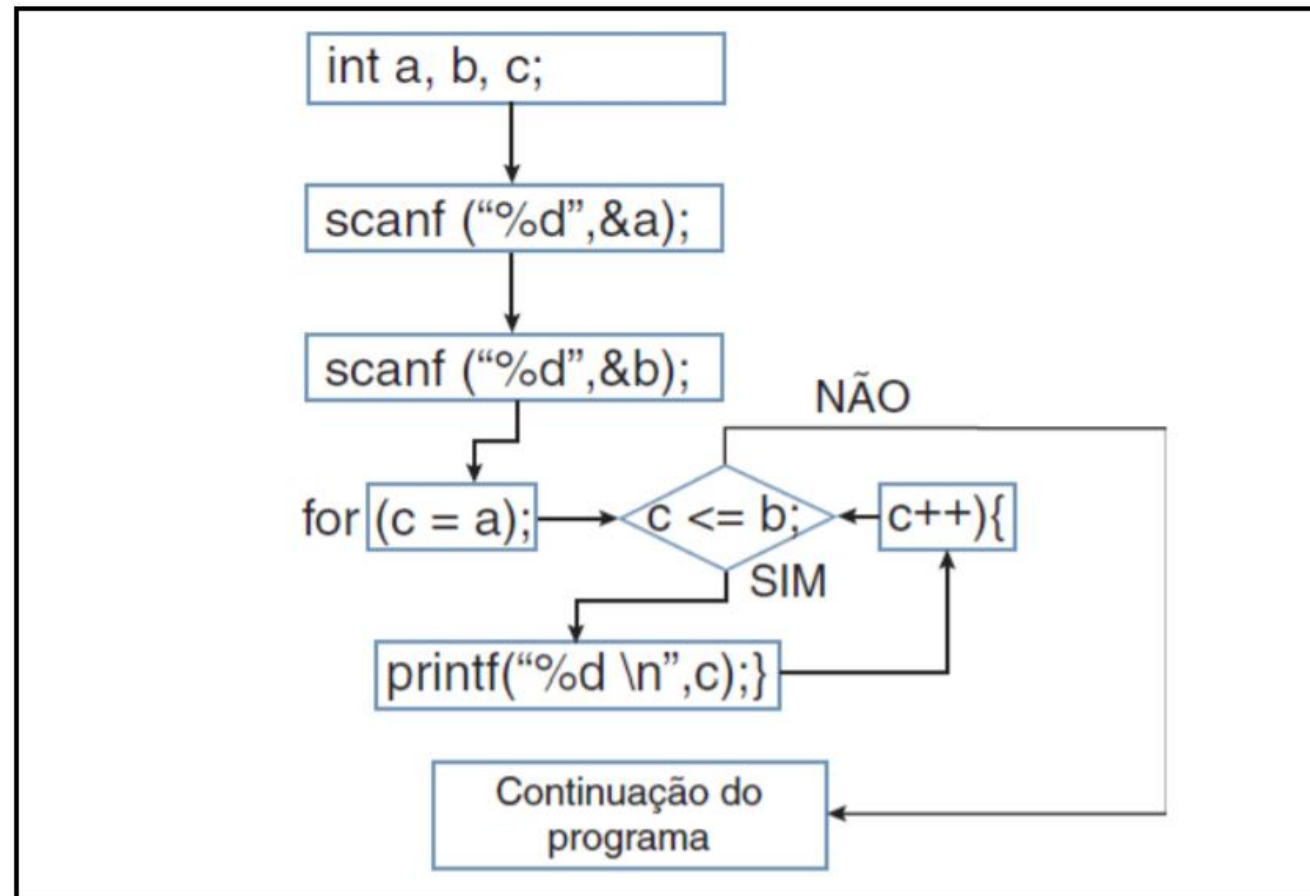
## EXEMPLO FOR

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int a,b,c;
    printf("Digite o valor de a: ");
    scanf("%d",&a);
    printf("Digite o valor de b: ");
    scanf("%d",&b);
    for(c = a; c <= b; c++){
        printf("%d \n",c);
    }

    return 0;
}
```



# NOVO EXEMPLO



# COMANDO FOR – PADRÃO ANSI / C99

- Por padrão o FOR permite que se declare uma variável em qualquer lugar do programa, inclusive dentro do comando FOR.

```
// ANSI C
int i;
for(i=1; i <= 10; i++)
    printf("%d\n",i);

// C99
for(int i=1; i <= 10; i++)
    printf("%d\n",i);
```

# COMANDO FOR

- Podemos omitir qualquer um de seus elementos:
  - Na inicialização, condição ou incremento.
  - Ex.: FOR sem inicialização.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int a,b,c;
    printf("Digite o valor de a: ");
    scanf("%d",&a);
    printf("Digite o valor de b: ");
    scanf("%d",&b);
    for (; a <= b; a++){
        printf("%d \n",a);
    }
    system("pause");
    return 0;
}
```

# COMANDO FOR

- **Cuidado: FOR sem condição!!!**
  - Omitir a condição cria um laço infinito;
  - Condição será sempre verdade;

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int a, b, c;
    printf("Digite o valor de a: ");
    scanf("%d", &a);
    printf("Digite o valor de b: ");
    scanf("%d", &b);
    //o comando for abaixo é um laço infinito
    for (c = a; ; c++) {
        printf("%d \n", c);
    }
    system("pause");
    return 0;
}
```

# COMANDO FOR

- **Cuidado: For sem incremento!!!**
  - Omitir o incremento cria um laço infinito;
  - Incremento pode ser feito nos comandos;

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int a,b,c;
    printf("Digite o valor de a: ");
    scanf("%d",&a);
    printf("Digite o valor de b: ");
    scanf("%d",&b);
    for (c = a; c <= b; ){
        printf("%d \n",c);
        c++;
    }
    system("pause");
    return 0;
}
```

# COMANDO FOR

- Podemos usar mais de uma inicialização ou incremento por vez. Basta separá-los por **vírgula**.

```
int i, j;  
for(i=1, j=10; i<=10; i++, j--)  
    printf("%d %d\n", i, j);
```

- Não se esqueça de usar o operador = quando trabalhar com outros incrementos.

```
for(int i=0; i<=10; i=i+2)  
    printf("%d\n", i);
```

## TREINAMENTO ...

- Escreva, usando FOR, um algoritmo para calcular a soma dos elementos de 1 a 10.

# RESOLUÇÃO

- Escreva, usando for, um algoritmo para calcular a soma dos elementos de 1 a 10.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
    int i, s = 0;
    for(i = 1; i <= 10; i++){
        s = s + i;
    }
    printf("Soma = %d \n",s);
    return 0;
}
```



# COMANDO BREAK

- Nós já vimos o uso para o comando break interrompendo os comandos switch, certo?

```
int num;  
scanf("%d", &num);  
switch(num) {  
    case 0: printf("Zero"); break;  
    case 1: printf("Um"); break;  
}
```

# COMANDO BREAK

- O comando break serve para:
  - Quebrar a execução de um comando (como no caso do switch);
  - Interromper a execução de qualquer loop (for, while ou do-while);
  - O comando break é utilizado para terminar de forma abrupta uma repetição. Por exemplo, se estivermos dentro de uma repetição e um determinado resultado ocorrer, o programa deverá sair da repetição e continuar na primeira linha seguinte a ela.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int cont;
7      for(cont=1; cont<=5; cont++)
8      {
9          if(cont==3)
10         {
11             break;
12         }
13         printf("%d, ", cont);
14     }
15     system("pause");
16     return 0;
17 }
18
19
```

## EXEMPLO DE BREAK

# COMANDO CONTINUE

- Diferente do comando break, só funciona dentro do loop;
  - “Pula” essa iteração do loop.
- Quando o comando continue é executado, os comandos restantes da repetição são ignorados. O programa volta a testar a condição do laço para saber se o mesmo deve ser executado novamente ou não;

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int cont;
7      for(cont=1;cont<=5;cont++)
8      {
9          if(cont==3)
10         {
11             continue;
12         }
13         printf("%d, ", cont);
14     }
15     system("pause");
16     return 0;
17 }
18
19
```

