

# Engenharia de Computação

## **Fundamentos de Programação**

### **Aula 10 – Estrutura de Repetição For**

**Prof. Muriel de Souza Godoi**  
[muriel@utfpr.edu.br](mailto:muriel@utfpr.edu.br)

# Laço de Repetição

- Lembrando...
  - É necessário quando se tem que repetir uma determinada instrução, que represente um comportamento padrão, várias vezes.
  - Com isso, escreve-se o comportamento padrão uma única vez e ela pode ser executada várias vezes.
  - As repetições podem ser controladas por uma condição.
  - Deve-se garantir que o laço acabe.

# Laço de Repetição - For

- Utilizado, principalmente, quando se sabe quantas vezes se quer que o comportamento padrão seja repetido
- O laço pode ser controlado por um contador
- Apesar do laço **while** ser suficiente para todos os tipos de repetição...
  - Em alguns casos, o **for** pode facilitar o código para fazer a mesma tarefa!



# Laço FOR

- **Sintaxe:**

```
for( <inicialização>; <condição>; <incremento> ){  
    //instruções que devem ser repetidas  
}  
//for
```

- **Exemplo:**

Incremento

```
for(contador = 0 ; contador < 5 ; contador++){  
    printf("Valor de c: %d", contador);  
}  
//for
```

Decremento

```
for(contador = 5 ; contador > 0 ; contador--){  
    printf("Valor de c: %d", contador);  
}  
//for
```

# Declarando variável no for

- Pode-se declarar a variável direto na inicialização para ser utilizada apenas no laço de repetição **for**

```
soma = 0;  
for ( int c = 0 ; c < 5 ; c++) {  
    soma = soma + c;  
} //for  
printf("Soma=%i", soma);
```



Declaração dentro do **for** foi incorporado à linguagem C na versão C99, para compilar utilize o parâmetro -std=c99 ou mais novo

# For com “várias variáveis”

- Pode-se declarar ou gerenciar mais variáveis e expressões no laço de repetição **for**

```
soma = 0;
for ( int c=0, d=5; c<5 && soma<6; c++, d=d-2) {
    soma = soma + c;
    printf("%i e %i\n", c, d);
} //for
printf("Soma=%i", soma);
```



Observe a diferença no uso da virgula e do ponto e vírgula

# Laço For Aninhado

- Quando um laço está dentro de outro, diz-se que o laço mais interno está aninhado.
- Laços aninhados propiciam um meio de resolver diversos problemas de programação.

```
for ( int x = 0 ; x < 10 ; x++){  
    for ( int y = 0 ; y < 10 ; y++){  
        printf("( %i,%i)",x,y);  
    }//for interno  
}//for externo
```



# Laço For Aninhado

- Para conseguir ver a saída mais claramente...

```
#include<stdio.h>
#include<stdlib.h>
#include<unistd.h> //necessario para usar o sleep()

int main() {
    for ( int x = 0 ; x < 10 ; x++){
        for ( int y = 0 ; y < 10 ; y++){
            printf("(%i,%i) ", x, y);
            fflush(stdout); //mostrar imediatamente na tela
            usleep(200000); //esperar 0.2 segundos
        } //for interno
        printf("\n");
    } //for externo
    return 0;
} //main
```



# Exercícios

- 1) Faça um algoritmo que exiba todos números de 1 a 100.
- 2) Faça um algoritmo que exiba todos números de 100 a 1.
- 3) Crie um algoritmo que exiba todos os números múltiplos de 5 no intervalo de 1 a 500.
- 4) Crie um algoritmo que exiba todos os números pares entre 240 e 730 inclusive.

# Exercícios

- 5) Faça um programa que calcule o fatorial de um número a ser digitado.
- 6) Faça um programa que receba 10 números e calcule o quadrado desse número (um de cada vez).
- 7) Faça um programa que mostre n elementos da sequência de Fibonacci.
  - O valor de n é passado para o programa.
  - O primeiro termo é 0, e o segundo termo é 1, os demais devem seguir a fórmula  $t_n = t_{n-1} + t_{n-2}$

# Exercícios

8) Utilizando a estrutura do laço for aninhado, crie um programa que exiba a tabuada de todos os números de 1 a 10 no seguinte formato:



Tabuada do 1:

1 x 1 = 1

1 x 2 = 2

1 x 3 = 3

...

Tabuada do 5:

5 x 1 = 5

5 x 2 = 10

5 x 3 = 15

...

# Exercícios

9) Desenvolva um programa que receba um numero tamanho e exiba um quadrado de tamanho tamanho utilizando o carácter #

**Exemplo:**



Digite o tamanho: 5

```
# # # # #  
# # # # #  
# # # # #  
# # # # #  
# # # # #
```



Para a exibição utilize apenas os comandos

```
printf("#");  
    e  
printf("\n");
```