

# Aula 3

## Estruturas de Repetição

**Responsável**

Prof. Seiji Isotani ([sisotani@icmc.usp.br](mailto:sisotani@icmc.usp.br))

- Comandos
  - For
  - While
- Exercícios

# Pseudocódigo para C

## Em Pseudocódigo

**PARA**  $i \leftarrow \text{valorInicial}$  **ATÉ**  $\text{valorFinal}$  **FAÇA**  
    Instrução\_1

---

## Em C

```
for (i = valorInicial; i <= valorFinal; i=i+1)  
    Instrução_1
```

;

## Pseudocódigo para C

### Em Pseudocódigo

**PARA**  $i \leftarrow \text{valorInicial}$  **ATÉ**  $\text{valorFinal}$  **FAÇA**

Instrução\_1

instrução\_2

---

### Em C

```
for (i = valorInicial; i <= valorFinal; i++){
```

```
    Instrução_1;
```

```
    instrução_2;
```

```
}
```

## Comando for

- Pode existir n comandos for aninhados
- Qualquer uma das três partes do laço “for” pode ser omitida
- Se as expressões de inicialização e de incremento forem omitidas, elas apenas serão desconsideradas
- Porém, se a expressão de teste for omitida, a mesma é considerada permanentemente verdadeira => `for( ; ; )`;
  - O que acontece nessa situação?

## Exemplo

Seu salário atual é de R\$5.000,00.  
O do seu colega é de R\$2.500,00.  
Suponha que vocês tenham 10%  
de aumento todo ano. Crie um  
algoritmo para calcular os salários  
daqui a 5 anos.

## Pseudocódigo:

Declare sal1, sal2

sal1 = 5000

sal2 = 2500

PARA i = 0 ATÉ 4 FAÇA

    sal1 = sal1\*1.1

    sal2 = sal2\*1.1

IMPRIME “sal1 e sal2”

## Em C:

```
float sal1, sal2;
```

```
sal1 = 5000;
```

```
sal2 = 2500 ;
```

```
for (i = 0; i <= 4; i++){
```

```
    sal1 = sal1*1.1
```

```
    sal2 = sal2*1.1
```

```
}
```

```
printf (“sal1 = %.2f e sal2 =  
%.2f”, sal1, sal2);
```

## Exercício

- Faça um programa que leia as 3 notas de cada um dos alunos de uma turma de 10 alunos, exibindo na tela a média simples de cada um destes alunos.

Exemplo:

>>Digite as 3 Notas do Aluno 1: 5.0 5.0 5.0

>>Media do Aluno 1: 5.00

>>Digite as 3 Notas do Aluno 2: 6.0 7.0 8.0

>>Media do Aluno 2: 7.00

>>...

>>Digite as 3 Notas do Aluno 10: 4.5 8.3 9.2

>>Media do Aluno 10: 7.33

## Exercício para entrega 1

- Um número natural é um número primo quando ele tem exatamente dois divisores distintos: o número um e ele mesmo.
- Faça um programa que leia um número fornecido e verifique se ele é primo ou não. O algoritmo deve ler até que o número informado seja -1. Exemplo:

>>Digite um número: 5

>>É Primo !!!

>>Digite um número: 20

>>Não é Primo !!!

>>Digite um número: -1

>>FIM

# Pseudocódigo para C - Comando While

## Em Pseudocódigo

**ENQUANTO** *condiçãoVerdadeira* **FAÇA**

Instrução\_1

Instrução\_2

---

## Em C

```
while (condiçãoVerdadeira) {
```

```
    Instrução_1;
```

```
    instrução_2;
```

```
}
```

# Comando While

```
int x, y;  
x = 1;  
y = 5;  
while(x < y) {  
    x = x+2;  
    y = y+1;  
}
```

## Comando While

- Se **condição** for **verdadeira** então executa cada um dos comandos do bloco subordinado ao **while**.
- Tenha atenção especial com o teste do comando while. As variáveis usadas no teste devem ter sido inicializadas **antes do teste**.
- Da mesma forma que o comando for, se a expressão de teste (condição) for omitida, a mesma é considerada permanentemente verdadeira

## Exercício Exemplo

- Seu salário atual é de R\$5.000,00. O do seu colega é de R\$2.500,00. Suponha que você tenha 10% de aumento todo ano e seu colega 20%. Crie um algoritmo que calcula em quanto tempo seu colega irá passar a ganhar mais que você.

## Comando - While

### Pseudocódigo:

Declare sal1, sal2

sal1 = 5000

sal2 = 2500

ENQUANTO sal1 > sal2 FAÇA

    sal1 = sal1\*1.1

    sal2 = sal2\*1.2

IMPRIME “sal1 e sal2”

### Em C:

```
float sal1, sal2;
```

```
sal1 = 5000;
```

```
sal2 = 2500 ;
```

```
while (sal1 > sa2){
```

```
    sal1 = sal1*1.1
```

```
    sal2 = sal2*1.2
```

```
}
```

```
printf (“sal1 = %.2f e sal2 =  
%.2f”, sal1, sal2);
```

## Exercício para Entrega 2

Faça um programa leia n números e para cada número imprima os números positivos antecedentes a ele na ordem decrescente. Exemplo:

```
>> Digite o valor de n: 2  
>> Digite um número: 10  
>> 9 8 7 6 5 4 3 2 1 0  
>> Digite um número: 7  
>> 6 5 4 3 2 1 0  
>> FIM
```

## Desafio

Começando com um valor inicial  $k$  (geralmente valendo 1), os babilônios geravam um novo valor de  $k$  de acordo com a regra:  $k = (k + N/k)/2$  para calcular a raiz quadrada de um número  $N$ .

A medida em que o processo é repetido, os novos valores de  $k$  se aproximam cada vez mais da raiz de  $N$ . Faça um algoritmo que leia o valor de  $N$  e exiba os números calculados com essa fórmula, verificando se eles realmente se aproximaram da raiz correta. Indique qual é o número  $x$  de iterações necessárias para que a raiz seja identificada com grau de confiança de pelo menos 2 casas decimais.