

# Aula 3

## Estruturas Condicionais

### Responsável

Prof. Armando Toda ([armando.toda@usp.br](mailto:armando.toda@usp.br))

### Monitores

Gustavo Barbosa

Emanuel de Oliveira

# Revisão

# Algumas Definições

- Leitura/Escrita (`#include <stdio.h>`)
  - Leitura de dados: `scanf()`;
  - Escrita de dados: `printf()`;
- Estrutura Condicional
  - Simples: `if-else`
  - Composta: `if-else if-else`
- Estrutura de Repetição
  - `for`
  - `do while`

## Exemplo - Pseudocódigo

Existem várias formas de escrever o pseudocódigo.  
Exemplos:

Multiplicação (n1, n2)

$m \leftarrow n1 * n2$

RETORNE m

FIM

ALGORITMO

DECLARE n1, n2, m

LEIA n1, n2

$m \leftarrow n1 * n2$

ESCREVA m

FIM

## Exemplo código C

### Código C

```
#include<stdio.h>
void main() {
    int n1, n2, m;
    scanf("%d %d", &n1, &n2);
    m = n1*n2;
    printf("\n %d", m);
}
```

### ALGORITMO

**DECLARE** n1, n2, m

**LEIA** n1, n2

**m** ← n1\*n2

**ESCREVA** m

**FIM**

# Algumas Definições

- Leitura/Escrita (#include <stdio.h>)
  - Leitura de dados: scanf();
  - Escrita de dados: printf();
- **Estrutura Condicional**
  - **Simplex: if-else**
  - **Composta: if-else if-else**
- Estrutura de Repetição
  - for
  - do while

# Estrutura Condicional Simples

- Condicionar a execução de determinado bloco ao resultado de uma **verificação**.

**X = 1**  
**(X >1 && X<4)**  
**(X>5 || X=5)**

```
if <condição> {  
    Instrução 1  
  
    ....  
    Instrução N  
}
```

# Estrutura Condicional Composta

```
IF <condição> {  
    Instrução 1  
    Instrução N  
} ELSE{ <condição>  
    Instrução 1  
    Instrução N  
}
```



# Estrutura Condicional

```
if(nota>=5){  
    printf("Aprovado");  
}else{  
    printf("Reprovado");  
}
```

```
if(nota>5){  
    printf("Aprovado");  
}  
if(nota<5){  
    printf("Reprovado");  
}
```

# Estrutura Condicional

```
if(nota>5) {  
    printf("Aprovado");  
    if(nota<5){  
        printf("Reprovado");  
    }  
}
```



# Estrutura Condicional

```
if(nota<5){  
    scanf("%f",&notaRec);  
    if(notaRec>5){  
        printf("Aprovado");  
    }else{  
        printf("Reprovado");  
    }else{  
        printf("Aprovado");  
    }  
}
```

# Estrutura Condicional

```
if(nota<5){  
    scanf("%f",&notaRec);  
    if(notaRec>5){  
        printf("Aprovado");  
    }else{  
        printf("Reprovado");  
    }  
}  
else{  
    printf("Aprovado");  
}  
}
```

## Exercício

- Faça um algoritmo para converter um peso expresso em libras para quilogramas
- $(1\text{Kg} = 1\text{Lb} / 2.2)$ 
  - Uma vez que o peso não pode ser um número negativo, o nosso programa não deve aceitar um número negativo como um peso válido.

## Estrutura Condicional Simples

- Condicionar a execução de determinado bloco ao resultado de uma **verificação**.

Composição de uma  
decisão

```
if(condição) {  
    Instrução 1  
  
    ....  
    Instrução N  
}
```

# Operadores lógicos

**E (&&)** - se e somente se os dois forem verdadeiros

X	Y	X&&Y
Verdadeiro	Verdadeiro	Verdadeiro
Falso	Verdadeiro	Falso
Verdadeiro	Falso	Falso
Falso	Falso	Falso

# Operadores lógicos

**OU (||)** - se e somente se pelo menos um for verdadeiro

X	Y	X  Y
Verdadeiro	Verdadeiro	Verdadeiro
Falso	Verdadeiro	Verdadeiro
Verdadeiro	Falso	Verdadeiro
Falso	Falso	Falso



# Operadores lógicos

**NÃO (!)** - inverte o valor do operando

X	!X
Verdadeiro	Falso
Falso	Verdadeiro

# Operadores relacionais

- Menor que ( $x < y$ )
- Menor ou igual a ( $x \leq y$ )
- Maior que ( $x > y$ )
- Maior ou igual a ( $x \geq y$ )
- Igual a ( $x == y$ )
- Diferente de ( $x \neq y$ )

## Exemplos

Dado **dois** inteiros crie um algoritmo para retornar o **maior** deles

## Estrutura Condicional Composta

**Entrada:** inteiros  $i$  e  $j$

**Saída:** um inteiro, o **maior** valor

```
if(i > j){
    printf("%d", i);
}else{
    printf("%d", j);
}
```

```
if( i > j){
    printf("%d", i);
}
if(i < j){
    printf("%d", j);
}
```

## Desafio - Exercício

Dado **três** inteiros crie um algoritmo para retornar o **menor** deles

## Resposta

LEIA n1, n2, n3

**SE** (n1 <= n2) **ENTAO**

**SE** (n1 <= n3) **ENTAO**

        IMPRIME n1

**SENAO**

        IMPRIME n3

**FIMSE**

**SENAO** // n1 > n2

**SE** (n2 <= n3) **ENTAO**

        IMPRIME n2

**SENAO**

        IMPRIME n3

**FIMSE**

**FIMSE**

## Exercício

Dado **três** inteiros crie um algoritmo para imprimi-los em ordem **crescente**

LEIA n1, n2, n3; DECLARE, x1, x2, x3

**SE** n1 ≤ n2 **ENTAO**

**SE** n1 ≤ n3 **ENTAO**

x1 ← n1

**SE** n2 ≤ n3 **ENTAO**

x2 ← n2; x3 ← n3

**SENAO**

x2 ← n3; x3 ← n2

**FIMSE**

**SENAO**

x1 ← n3; x2 ← n1; x3 ← n2

**FIMSE**

**SENAO** // n1 > n2

**SE** n2 ≤ n3 **ENTAO**

x1 ← n2

**SE** n1 ≤ n3 **ENTAO**

x2 ← n1; x3 ← n3

**SENAO**

x2 ← n3; x3 ← n1

**FIMSE**

**SENAO**

x1 ← n3; x2 ← n2; x3 ← n1