

Professor: Fábio Ferreira de Assis

# Computação

Uma imersão ao mundo binário



USCS  
UNIVERSIDADE DE SÃO CAETANO DO SUL

01000011  
01101111  
01101101  
01110000  
01110101  
01110100  
01100001  
11100111  
11100011  
01101111

## Agenda Aula 1-2-3: REVISÃO

Conceitos Básicos de Programação (Algoritmos e Lógica de Programação)

Estruturas Básicas e Estruturas de Decisão

Estruturas de Repetição

Exercícios

# Conceitos Básicos de Programação

## Revisão

Objetivos:

Compreender os componentes básicos de um programa ...

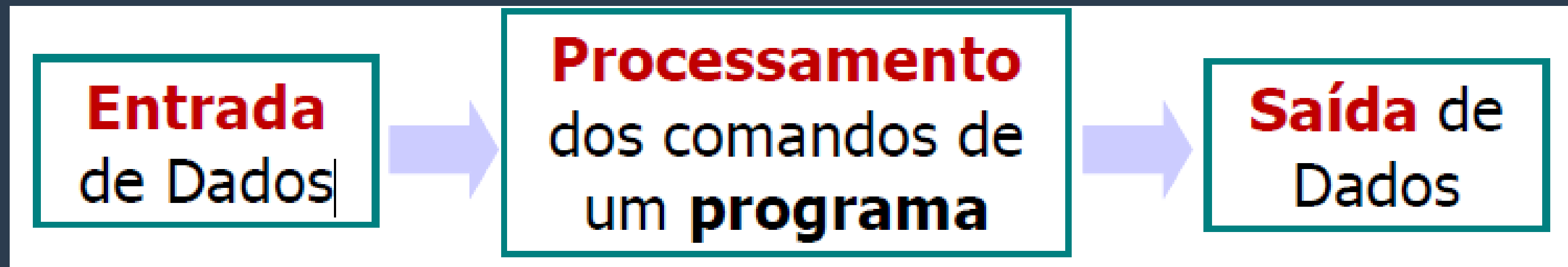
Objetivos específicos:

Aprender e praticar a escrita de PROGRAMAS de computador para resolver problemas.

## Conceitos Básicos

### PROGRAMA

Podemos definir programa, como uma sequência de instruções (comandos) que, a partir de dados inseridos, obtêm um resultado que será disponibilizado por algum dispositivo de saída



## Conceitos Básicos

### EXPRESSÕES

Operador de atribuição =

Exemplo: variável n1 = 20;

Exemplo: variável n2 = 5;

Operador de comparação ==

Exemplo: n1 == n2

Retorna 1, se o conteúdo de n1 for igual o conteúdo da variável n2

Retorna 0, se n1 diferente de n2

Aritméticas (+ - \* / ^ sqrt mod)

Lógicas (.NÃO. .E. .OU.)

Relacionais (> < >= <= = <>)

Literais (concatenação/cópia)

## Conceitos Básicos

### OPERADORES RELACIONAIS

<u>Operador</u>	<u>Comparação</u>	<u>Precedência (ordem)</u>
<	Menor	1ª
<=	Menor ou igual	1ª
>	Maior	1ª
>=	Maior ou igual	1ª
=	Igual	2ª
<>	Diferente	2ª

## Conceitos Básicos

### PRIORIDADE ENTRE OPERADORES

<u>Operação</u>	<u>Precedência (ordem)</u>
Operadores aritméticos/literais	1ª
Operadores relacionais	2ª
Operadores lógicos	3ª

## ***Conceitos Básicos***

### **PRIORIDADE ENTRE OPERADORES**

Operadores Aritméticos **+**, **-**, **\***, **/**, **\** (divisão inteira), **MOD**(resto da divisão inteira) **^** (potenciação)

Operadores Relacionais **==**, **<=**, **>=**, **<**, **>**

Operadores Lógicos **ou**, **nao**, **e**, **xou**

Entrada de Dados **leia**

Saída de Dados **escreva**

\* Caso utilize a biblioteca (Matemática --> mat), pode-se trabalhar com funções prontas para cálculo com potência, raiz quadrada, outros.



## ***Conceitos Básicos***

### **PRIORIDADE ENTRE OPERADORES**

Quando várias operações ocorrem em uma expressão, cada parte é avaliada e resolvida em uma ordem predeterminada chamada precedência do operador.

Quando as expressões contêm operadores de mais de uma categoria, os operadores aritméticos são avaliados primeiro, os operadores de comparação são avaliados em seguida, e os operadores lógicos são avaliados por último.

Operadores de comparação todos têm precedência; ou seja, eles são avaliados na ordem da esquerda para a direita em que aparecem.

## ***Conceitos Básicos***

### ***PRIORIDADE ENTRE OPERADORES***

Quando a multiplicação e a divisão ocorrem em conjunto em uma expressão, cada operação é avaliada conforme ela ocorre da esquerda para a direita.

Quando a adição e a subtração ocorrem juntas em uma expressão, cada operação é avaliada em ordem de aparência da esquerda para a direita.

# Conceitos Básicos

## PRIORIDADE ENTRE OPERADORES

Operadores aritméticos e lógicos são avaliados na seguinte ordem de precedência:

Aritmética	Comparação	Lógica
Exponenciação (^)	Igualdade (=)	Not
Negação (-)	Inequality (< >)	And
Multiplicação e divisão (*, /)	Menor que (<)	Or
Divisão de inteiro \()	Maior que (>)	Xor
Aritmética de módulo Mod()	Menor que ou igual a (<=)	Eqv
Adição e subtração (+, -)	Maior ou igual a (>=)	Imp
Concatenação de cadeia & de caracteres ()	Like, é	

## ***Conceitos Básicos***

### ***ESTRUTURA SEQUÊNCIAL***

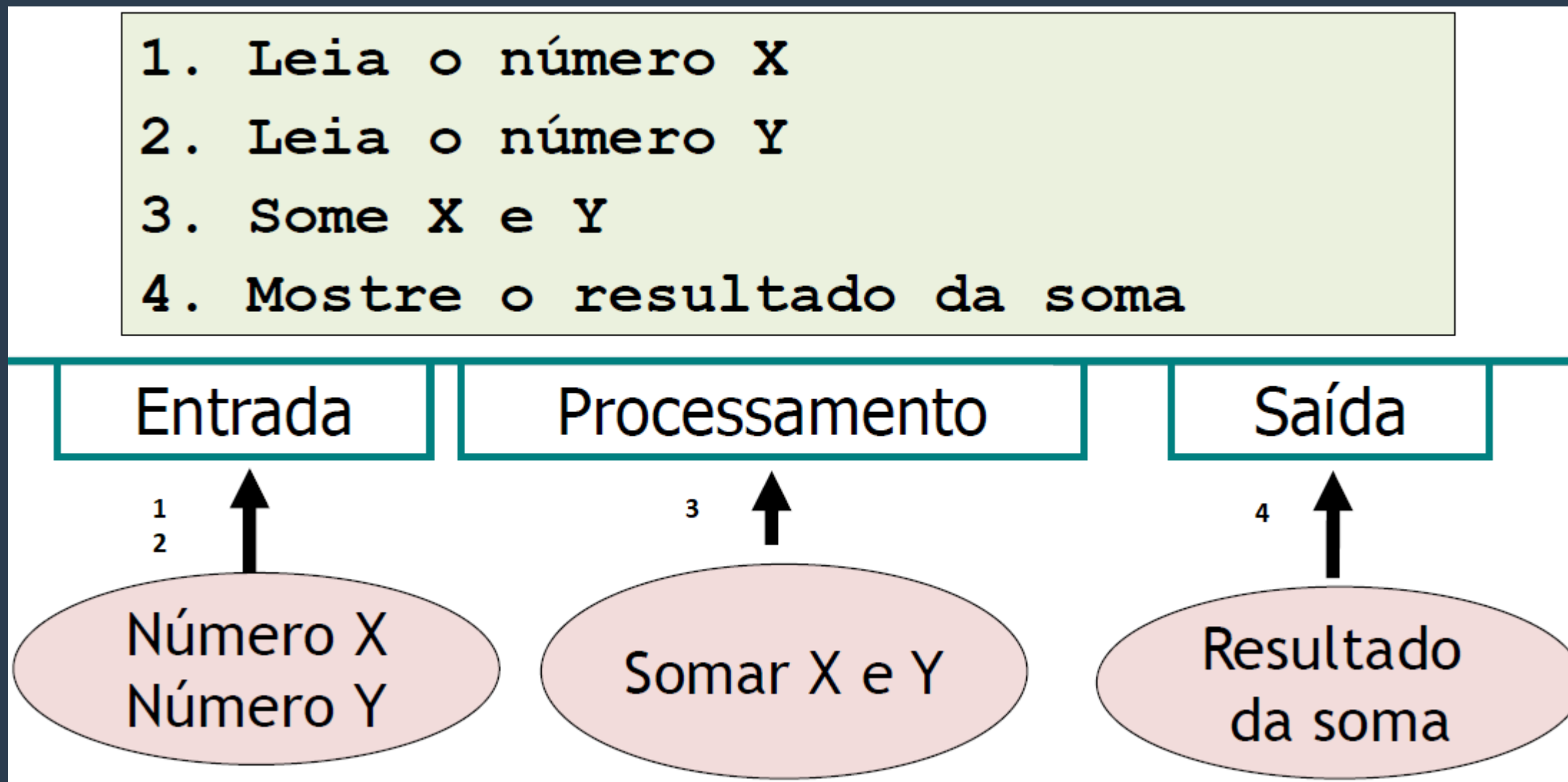
Uma estrutura sequencial é um conjunto de comandos que serão executados em uma sequência linear, de cima para baixo.

Os comandos serão executados na mesma ordem em que foram escritos. Até o momento todos os exemplos e exercícios foram resolvidos com estruturas sequenciais.

## Exemplo 1:

### *EXEMPLO DE ALGORITMO*

Algoritmo (Programa) que soma dois números e exibe o resultado na tela



## Exemplo 2:

### EXEMPLO DE ALGORITMO

Algoritmo (Programa) que soma dois números e exibe o resultado na tela

```
1. programa
2. {
3.     funcao inicio()
4.     {
5.         inteiro a, b, soma
6.
7.         escreva("Digite o valor de a: ")
8.         leia(a)
9.         escreva("Digite o valor de b: ")
10.        leia(b)
11.
12.        soma = a + b // Soma os dois valores
13.
14.        escreva("\nA soma dos números é igual a: ", soma) // Exibe o resultado
15.
16.    }
17. }
```

Declaração de Variáveis

Entrada de Dados

Entrada de Dados

Processamento

Saída de Dados

```
1  #include <stdio.h>
2
3  int main()
4  {
5      int a, b, soma;
6
7      printf("Digite o valor de a: ");
8      scanf("%d", &a);
9
10     printf("Digite o valor de b: ");
11     scanf("%d", &b);
12
13     soma = a + b;
14
15     printf("\nA soma dos números é igual a: %d", soma);
16
17     return 0;
18 }
```

## Exemplo 4:

### *EXEMPLO DE ALGORITMO*

Algoritmo (Programa) que calcula o resto da divisão de um número por 2 e exibe o resultado na tela

```
programa
{
    funcao inicio()
    {
        inteiro resto, numero

        escreva("Digite um valor: ")
        leia(numero)

        // Calcula o resto da divisão do valor por 2
        resto = numero % 2

        escreva("\nO resto (mod) da divisão por 2 = ", resto, "\n")
    }
}
```

/> source code

```
1  #include <stdio.h>
2
3  int main(void) {
4      int numero, resto;
5
6      printf("Digite um valor: ");
7      scanf("%d", &numero);
8
9      resto = numero % 2;
10
11     printf("O resto (mod) da divisão por 2 = %d", resto);
12
13     return 0;
14 }
```

## Exemplo 5:

### **EXEMPLO DE ALGORITMO**

Programa que multiplica dois números e exibe o resultado na tela.

<pre>programa {     funcao inicio() {         inteiro a, b, soma, mult         escreva("Digite o primeiro número: ")         leia(a)          escreva("\nDigite o segundo número: ")         leia(b)          mult = a * b // Multiplica os dois valores          // Exibe o resultado da multiplicação         escreva("\nA multiplicação dos números é : ", mult)     } }</pre>	<pre>// programa em C  #include &lt;stdio.h&gt; int main(void) {     int a, b, soma, mult;     printf("Digite o primeiro número: ");     scanf("%d", &amp;a);      printf("\nDigite o segundo número: ");     scanf("%d", &amp;b);      mult = a * b;      printf("\nA multiplicação dos números é: %d", mult);      return 0; }</pre>
---	--



## Exemplo com texto

Lendo uma string em C  
Usando scanf

A função scanf permite fazer leitura de strings usando %s.

Em relação ao uso de scanf para armazenar string devemos observar duas coisas:

A função scanf realiza a leitura até encontrar um espaço, depois encerra a leitura e coloca o caracter terminador \0.

A variável que vai armazenar a string não necessita ser precedida por &.

Exemplo de utilização do scanf:

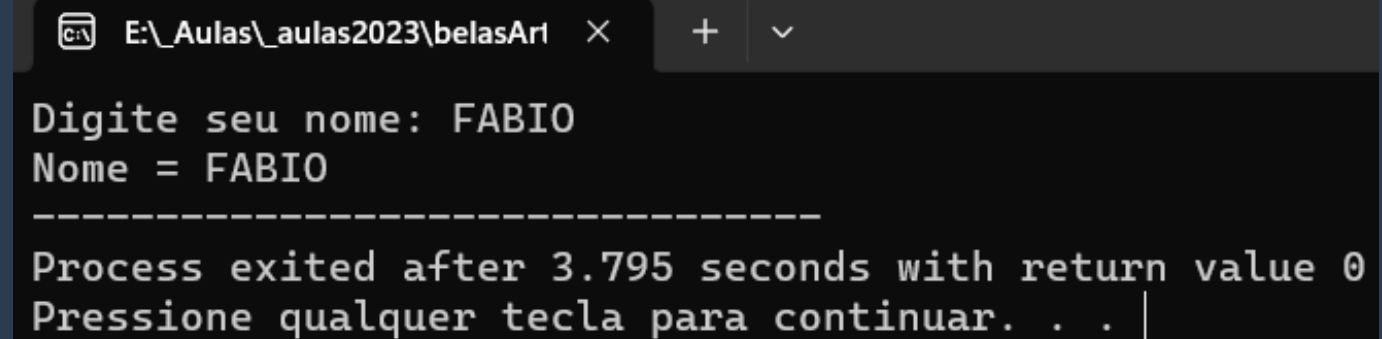
```
scanf("%s",nome);
```

## Exemplo com texto

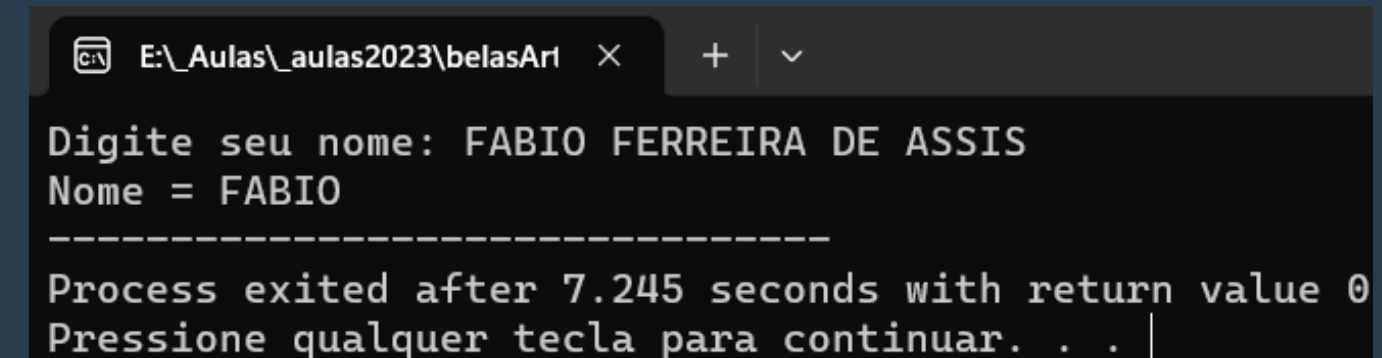
Exemplo: Usando scanf para receber uma string.

```
1.  #include<stdio.h>
2.  #include<conio.h>
3.
4.  int main(void)
5.  {
6.      char nome[61];
7.
8.      printf("Digite seu nome: ");
9.      scanf("%s",nome);
10.
11.     printf("O nome armazenado foi: %s", nome);
12.
13.
14.     return 0;
15. }
```

### EXEMPLO DE SAÍDA NO CONSOLE



```
E:\Aulas\aulas2023\belasArt x + v
Digite seu nome: FABIO
Nome = FABIO
-----
Process exited after 3.795 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```



```
E:\Aulas\aulas2023\belasArt x + v
Digite seu nome: FABIO FERREIRA DE ASSIS
Nome = FABIO
-----
Process exited after 7.245 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

## Exemplo com texto

Em C declaramos um vetor de caracteres exemplo: `char str[30]`, logo podemos fazer a leitura com o comando `scanf()`; conforme abaixo:

### Exemplo

```
scanf("%30[^\\n]*c",str);
```

^    ^    ^    ^-----String a ser pega  
|    |    |-----Caractere de controle  
|    |-----Ignora NewLine  
|-----Pega 29 caracteres

### </> source code

```
1  #include <stdio.h>
2
3  int main(void) {
4      char nome[30];
5      printf("Digite seu nome: ");
6      scanf("%30[^\\n]*c", nome);
7
8      printf("\\nNOME: %s", nome);
9      return 0;
10 }
```

Quando trabalhamos com *strings* não precisamos colocar o símbolo `&`, antes da variável

## Exemplo com texto

### EXEMPLO DE SAÍDA NO CONSOLE

```
*] exemploChar.cpp
1  #include <stdio.h>
2
3  int main(){
4      char nome[50];
5
6      printf("Digite seu nome: ");
7      scanf("%50[^\n]*c", nome);
8
9      printf("Nome = %s", nome);
10
11     return 0;
12 }
```

```
E:\_Aulas\_aulas2023\belasArt
Digite seu nome: FABIO FERREIRA DE ASSIS
Nome = FABIO FERREIRA DE ASSIS
-----
Process exited after 6.122 seconds with return value 0
Pressione qualquer tecla para continuar. . . |
```

## Exemplo com texto

### Função gets – (get string)

Esta função armazena tudo que foi digitado, inclusive os espaços, até que a tecla ENTER seja pressionada.

Exemplo:

```
printf (“Digite se nome: ”);  
gets(nome);
```

```
exemploChar.cpp ×  
1  #include <stdio.h>  
2  
3  int main(){  
4      char nome[50];  
5  
6      printf("Digite seu nome: ");  
7      gets(nome);  
8  
9      printf("Nome = %s", nome);  
10  
11     return 0;  
12 }
```

### EXEMPLO DE SAÍDA NO CONSOLE

```
E:\_Aulas\_aulas2023\belasArt × + v  
Digite seu nome: FABIO FERREIRA DE ASSIS  
Nome = FABIO FERREIRA DE ASSIS  
-----  
Process exited after 5.191 seconds with return value 0  
Pressione qualquer tecla para continuar. . . |
```

## Exemplo com texto

### Função gets – (get string)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(void) {
5      char palavra[20] = {"Hello World!"};
6
7      gets(palavra);
8
9      printf("\n%s", palavra);
10     return 0;
11 }
```

## Exemplo 1.2:

### EXEMPLO DE ALGORITMO

Algoritmo (Programa) que soma dois números exibe o resultado na tela

```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro numeroX=0, numeroY=0, soma=0
6         escreva("\nDIGITE o primeiro NÚMERO \n")
7         leia(numeroX)
8
9         escreva("\nDIGITE o segundo NÚMERO \n")
10        leia(numeroY)
11
12        soma = numeroX + numeroY
13
14        escreva("\nSOMA = ", soma, "\n")
15    }
16 }
```

```
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main() {
5     setlocale(LC_ALL, "Portuguese");
6
7     int numeroX=0, numeroY=0, soma;
8
9     printf("\nDigite o primeiro número: ");
10    scanf("%d", &numeroX);
11
12    printf("\nDigite o segundo número: ");
13    scanf("%d", &numeroY);
14
15    soma = numeroX + numeroY;
16
17    printf("SOMA = %d", soma);
18
19    return 0;
20 }
```

## Exemplo 1.3:

### EXEMPLO DE ALGORITMO

Algoritmo (Programa) que soma dois números exibe o resultado na tela

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int numeroX=0, numeroY=0;
8
9      printf("\nDigite o primeiro número: ");
10     scanf("%d", &numeroX);
11
12     printf("\nDigite o segundo número: ");
13     scanf("%d", &numeroY);
14
15     printf("\nSOMA = %d", numeroX + numeroY);
16
17     return 0;
18 }
```

```
E:\_Aulas\_aulas2023\belasArt
Digite o primeiro número: 9
Digite o segundo número: 1
SOMA = 10
Process returned 0 (0x0)
Press any key to continue.
```



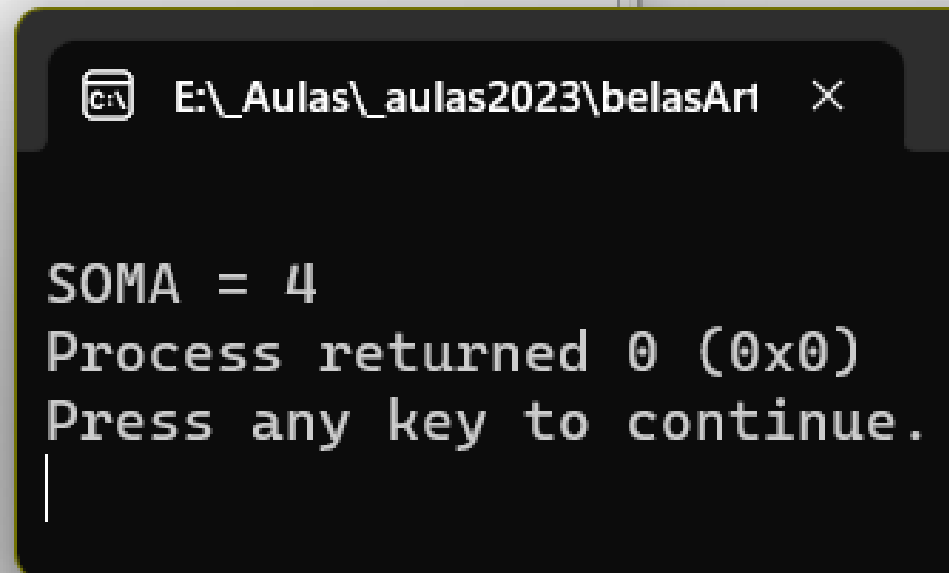
## Exemplo 2:

### *EXEMPLO DE ALGORITMO*

Algoritmo (Programa) que soma dois números exibe o resultado na tela

```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro a=1, b=3, soma
6
7         soma = a + b
8
9         escreva("\n\nSOMA = ", soma)
10    }
11 }
```

```
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main() {
5     setlocale(LC_ALL, "Portuguese");
6
7     int a=1, b=3, soma=0;
8
9     soma = a+b;
10
11     printf("\nSOMA = %d", soma);
12
13     return 0;
14 }
```



```
E:\Aulas\aulas2023\belasArt X
SOMA = 4
Process returned 0 (0x0)
Press any key to continue.
```

ALGORITMO

## Exemplo 2.2:

### EXEMPLO DE ALGORITMO

Algoritmo (Programa) que soma dois números exibe o resultado na tela

```
1 programa
2 {
3
4     funcao inicio()
5     {
6         inteiro a = 2, b = 3
7         //a operação de soma foi efetuada dentro da função escreva
8
9         escreva("SOMA = ", a+b, "\n")
10    }
11 }
```

SOMA = 4  
Programa finalizado. Tempo de execução: 35 milissegundos

```
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main() {
5     setlocale(LC_ALL, "Portuguese");
6
7     int a=1, b=3;
8
9     printf("\nSOMA = %d", a+b);
10
11     return 0;
12 }
```

E:\Aulas\aulas2023\belasArt × + ▾

SOMA = 4  
Process returned 0 (0x0) execution time : 0.051 s  
Press any key to continue.

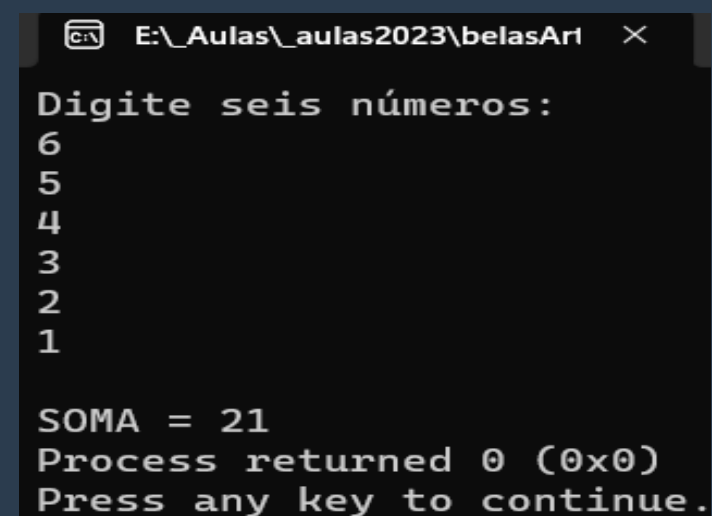
## Exemplo 2.3:

### EXEMPLO DE ALGORITMO

Algoritmo (Programa) que soma seis números exibe o resultado na tela

```
1 //programa que soma seis números
2 programa
3 {
4     funcao inicio()
5     {
6         inteiro n1, n2, n3, n4, n5, n6, soma
7         escreva("Digite seis números: ")
8
9         leia(n1)
10        leia(n2)
11        leia(n3)
12        leia(n4)
13        leia(n5)
14        leia(n6)
15
16        soma = n1+n2+n3+n4+n5+n6
17
18        escreva("Soma dos valores = ", soma)
19    }
20 }
```

```
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main(){
5     setlocale(LC_ALL, "Portuguese");
6
7     int n1=0, n2=0, n3=0, n4=0, n5=0, n6=0, soma=0;
8
9     printf("Digite seis números:\n");
10
11     scanf("%d %d %d %d %d %d", &n1, &n2, &n3, &n4, &n5, &n6);
12
13     soma = n1+n2+n3+n4+n5+n6;
14     printf("\nSOMA = %d", soma);
15
16     return 0;
17 }
```



```
E:\Aulas\aulas2023\belasArt
Digite seis números:
6
5
4
3
2
1

SOMA = 21
Process returned 0 (0x0)
Press any key to continue.
```

## Exemplo 2.4:

### EXEMPLO DE ALGORITMO

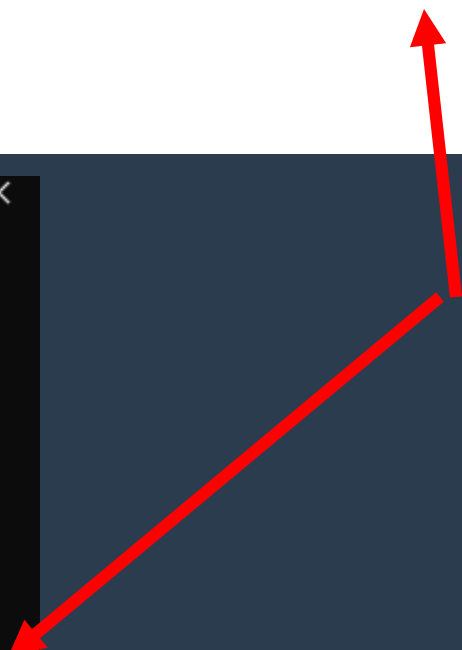
Algoritmo (Programa) que calcula a média de seis números exibe o resultado na tela

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int n1=0, n2=0, n3=0, n4=0, n5=0, n6=0;
8      float media=0.0;
9
10     printf("Digite seis números:\n");
11
12     scanf("%d %d %d %d %d %d", &n1, &n2, &n3, &n4, &n5, &n6);
13
14     media = (n1+n2+n3+n4+n5+n6)/6.0;
15     printf("\nMédia dos valores = %f", media);
16
17     return 0;
18 }
```

```
E:\Aulas\aulas2023\belasArt x +
Digite seis números:
6
5
4
3
2
1
Média dos valores = 3,500000
Process returned 0 (0x0)
Press any key to continue.
Fábio Ferre
```

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int n1=0, n2=0, n3=0, n4=0, n5=0, n6=0;
8      float media=0.0;
9
10     printf("Digite seis números:\n");
11
12     scanf("%d %d %d %d %d %d", &n1, &n2, &n3, &n4, &n5, &n6);
13
14     media = (n1+n2+n3+n4+n5+n6)/6.0;
15     printf("\nMédia dos valores = %.2f", media);
16
17     return 0;
18 }
```

```
E:\Aulas\aulas2023\belasArt x
Digite seis números:
6
5
4
3
2
1
Média dos valores = 3,50
```



## Exemplo 2.4:

### EXEMPLO DE ALGORITMO

Algoritmo (Programa) que calcula a média ponderada de seis números exibe o resultado na tela

Nota1 (peso 5)

Nota2 (peso 3)

Nota3 (peso 2)

Nota4 (peso 3)

Nota5 (peso 2)

Nota6 (peso 4)

Faça em C

```
1 //programa que calcula a média de seis números
2 //ponderada por pesos
3 programa
4 {
5     funcao inicio()
6     {
7         inteiro n1, n2, n3, n4, n5, n6
8         real media
9         escreva("Digite seis números: ")
10
11         leia(n1)
12         leia(n2)
13         leia(n3)
14         leia(n4)
15         leia(n5)
16         leia(n6)
17
18         media = ((n1*5)+(n2*3)+(n3*2)+(n4*3)+(n5*2)+(n6*4))/19.0
19
20         escreva("Média dos valores = ", media)
21     }
22 }
```

## Exemplo 3.2:

### EXEMPLO DE ALGORITMO

Algoritmo (Programa) que soma dois números inteiros digitados pelo usuário e mostra o resultado na tela, conforme exemplo:  $\text{soma} = \text{valor1} + \text{valor2} = \text{resultado}$

```
1 programa
2 {
3     funcao inicio()//função principal
4     {
5         inteiro nX=0, nY=0, soma=0
6         escreva("Digite o número 1: ")
7         leia(nX)
8         limpa()
9         escreva("Digite o número 2: ")
10        leia(nY)
11        limpa()
12
13        soma = nX + nY
14        escreva("soma = ", nX, " + ", nY, " = ", soma)
15    }
16 }
```

```
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main(){
5     setlocale(LC_ALL, "Portuguese");
6
7     int nX=0, nY=0, soma=0;
8
9     printf("Digite o número 1: ");
10    scanf("%d", &nX);
11    printf("Digite o número 2: ");
12    scanf("%d", &nY);
13
14    soma = nX+nY;
15
16    printf("\nsoma = %d + %d = %d\n", nX, nY, soma);
17
18    return 0;
19 }
```

```
E:\Aulas\aulas2023\belasArt X
Digite o número 1: 5
Digite o número 2: 2

soma = 5 + 2 = 7

Process returned 0 (0x0)
Press any key to continue.
```

## Exemplo 3.3:

### ***EXEMPLO DE ALGORITMO***

Algoritmo (Programa) que soma dois números inteiros e mostra o resultado na tela

Conforme exemplo: soma = valor1 + valor2 = resultado

```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro a=3, b=3, soma=0
6         soma = a + b
7         escreva("soma = ", a, " + ", b, " = ", soma)
8     }
9 }
```



## Exemplo 3.5

### EXEMPLO DE ALGORITMO

Algoritmo (Programa) que soma dois números reais e mostra o resultado na tela

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      float nX=0, nY=0, soma=0;
8
9      printf("Digite o número 1: ");
10     scanf("%f", &nX);
11     printf("Digite o número 2: ");
12     scanf("%f", &nY);
13
14     soma = nX+nY;
15
16     printf("\nsoma = %f + %f = %f\n", nX, nY, soma);
17
18     return 0;
19 }
```

```
E:\Aulas\aulas2023\belasArt
Digite o número 1: 5
Digite o número 2: 2

soma = 5,000000 + 2,000000 = 7,000000
```

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      float nX=0, nY=0, soma=0;
8
9      printf("Digite o número 1: ");
10     scanf("%f", &nX);
11     printf("Digite o número 2: ");
12     scanf("%f", &nY);
13
14     soma = nX+nY;
15
16     printf("\nsoma = %.1f + %.1f = %.2f\n", nX, nY, soma);
17
18     return 0;
19 }
```

```
E:\Aulas\aulas2023\belasArt
Digite o número 1: 5
Digite o número 2: 2

soma = 5,0 + 2,0 = 7,00
```

Neste exemplo, utilizamos %.1f, para formatar a saída das variáveis nX e nY em uma casa decimal. E %.2f para a saída da variável soma apresentar duas casas decimais.



## Exemplo 4.1:

### ***EXEMPLO DE COMPARAÇÃO***

Programa que recebe do usuário um nome digitado e exibe o resultado na tela.

```
programa
{
    funcao inicio()
    {
        cadeia nome
        escreva("Digite seu nome: ")
        leia(nome)

        // Exibe o resultado
        escreva("\nNome: ", nome)
    }
}
```

## Exemplo 4.2:

### ***EXEMPLO DE COMPARAÇÃO***

Programa que recebe do usuário um nome digitado e exibe o resultado na tela.

*programa*

```
{  
    funcao inicio()  
    {  
        cadeia nome  
        escreva("Digite seu nome: ")  
        leia(nome)  
  
        // Exibe o resultado  
        escreva("\nNome: ", nome)  
    }  
}
```

*// programa em C*

```
#include <stdio.h>  
int main(void)  
{  
    char nome[10];  
    //scanf("%s", nome);  
  
    gets(nome);  
  
    printf("nome: %s", nome);  
  
    return 0;  
}
```

## Exemplo 7.1:

### ***EXEMPLO DE ALGORITMO***

Programa que multiplica dois números e exibe o resultado na tela.

```
programa
{
    funcao inicio()
    {
        inteiro a, b, soma, mult
        escreva("Digite o primeiro número: ")
        leia(a)

        escreva("\nDigite o segundo número: ")
        leia(b)

        mult = a * b // Multiplica os dois valores

        // Exibe o resultado da multiplicação
        escreva("\nA multiplicação dos números é : ", mult)
    }
}
```

## Exemplo 7.2:

### **EXEMPLO DE ALGORITMO**

Programa que multiplica dois números e exibe o resultado na tela.

<i>programa</i>	<i>// programa em C</i>
{	
<i>funcao inicio()</i>	<i>#include &lt;stdio.h&gt;</i>
{	<i>int main(void)</i>
<i>inteiro a, b, soma, mult</i>	{
<i>escreva("Digite o primeiro número: ")</i>	<i>int a, b, soma, mult;</i>
<i>leia(a)</i>	<i>printf("Digite o primeiro número: ");</i>
	<i>scanf("%d", &amp;a);</i>
<i>escreva("\nDigite o segundo número: ")</i>	
<i>leia(b)</i>	<i>printf("\nDigite o segundo número: ");</i>
	<i>scanf("%d", &amp;b);</i>
<i>mult = a * b // Multiplica os dois valores</i>	
	<i>mult = a * b;</i>
<i>// Exibe o resultado da multiplicação</i>	
<i>escreva("\nA multiplicação dos números é : ", mult)</i>	<i>printf("\nA multiplicação dos números é: %d", mult);</i>
}	
}	<i>return 0;</i>
	}

# Exercícios de Revisão:

Exemplo em C

Para imprimir um inteiro com *printf* usa-se o símbolo %d

```
// Impressão de Variáveis Inteiras
#include <stdio.h>
#include <conio.h>    // necessário para as funções clrscr e getch

void main ()
{
    int Contador;
    int NroDeFilhos;

    clrscr();        // Limpa a tela

    Contador = 10;
    printf("Valor da Variável: %d\n", Contador);           // No momento da execução sinal %d vai
                                                            // ser substituído pelo valor da
                                                            // variável Contador

    NroDeFilhos = 3;
    printf("Maria tem %d filhos", NroDeFilhos); // o inteiro pode ficar no meio da string

    getch();        // espera que o usuário pressione uma tecla
}
```

# Exercícios de Revisão:

## Exemplo em C

```
1  #include <stdio.h>
2  #include <conio.h>
3  #include <stdlib.h>
4
5  int main() {
6      char nome[50];
7      int qtde=0;
8      float total=0, valor = 0;
9
10     printf("Digite um produto: ");
11     //scanf("%50[^\n]%*c", nome);
12     gets(nome);
13
14     printf("Digite o valor de %s: ", nome);
15     scanf("%f", &valor);
16
17     printf("Digite a quantidade de %s: ", nome);
18     scanf("%d", &qtde);
19
20     //clrscr();
21
22     total = valor * qtde;
23
24     printf("Produto %s Total = R$ %.2f", nome, total);
25
26     if(total >= 1000){
27         float imposto = total * 0.15;
28         printf("\nTêm que pagar imposto de R$ %.2f total Final R$ %.2f", imposto, total+imposto);
29     }
30
31     getch();
32     return 0;
33 }
```

## Exemplo com texto

Em C declaramos um vetor de caracteres exemplo: `char str[30]`, logo podemos fazer a leitura com o comando `scanf()`; conforme abaixo:

### Exemplo

```
scanf("%30[^\\n]*c",str);
```

^     ^   ^     ^-----String a ser pega  
|     |   |-----Caractere de controle  
|     |-----Ignora NewLine  
|-----Pega 29 caracteres

### </> source code

```
1  #include <stdio.h>
2
3  int main(void) {
4      char nome[30];
5      printf("Digite seu nome: ");
6      scanf("%30[^\\n]*c", nome);
7
8      printf("\\nNOME: %s", nome);
9      return 0;
10 }
```

Quando trabalhamos com *strings* não precisamos colocar o símbolo `&`, antes da variável

Exercícios de Revisão:

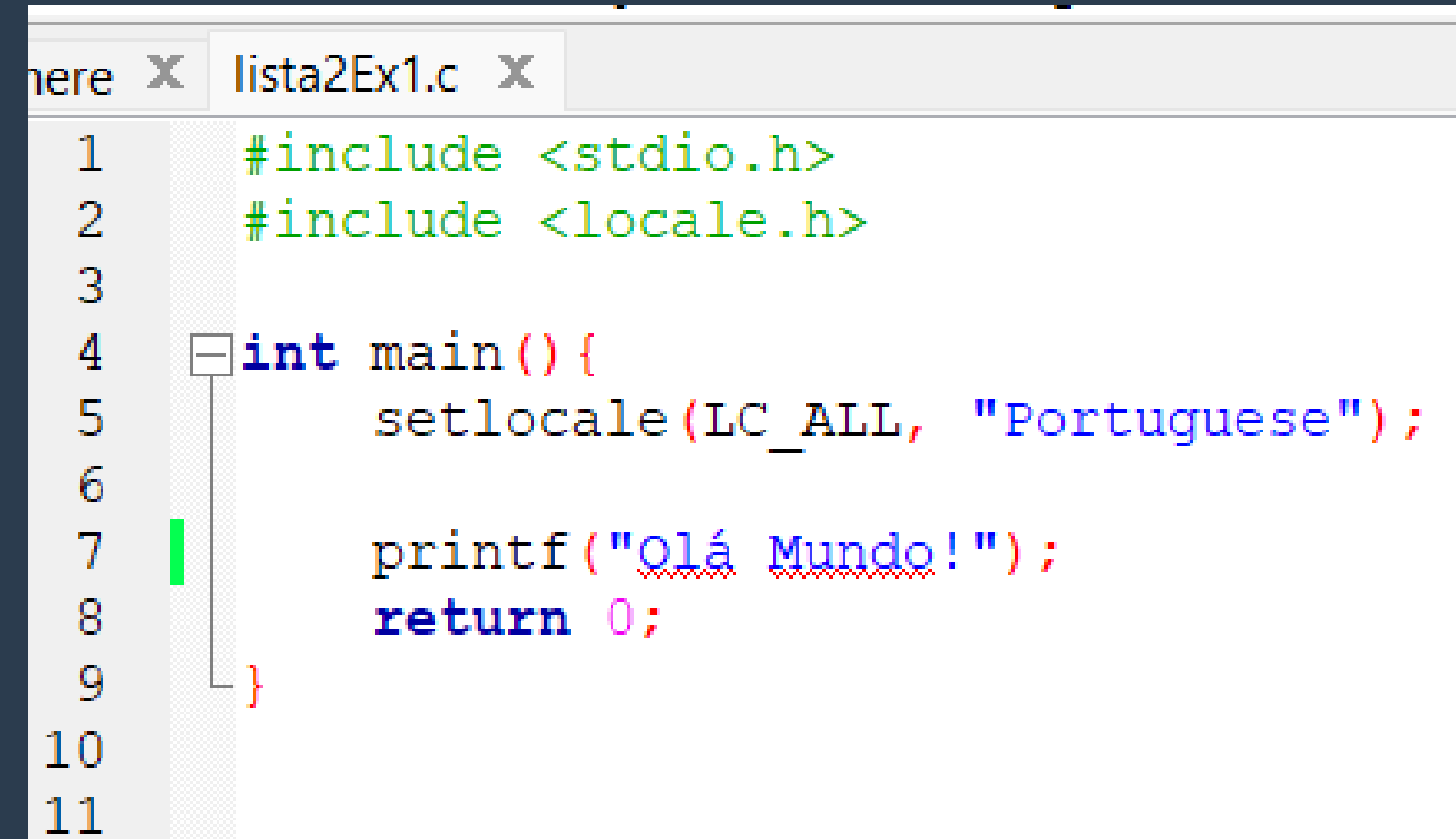
# RESOLUÇÃO DE EXERCÍCIOS DE REVISÃO DE CONTEÚDO



## Exercícios de Revisão:

1) Faça um programa que exibe a mensagem “Olá Mundo!”.

```
programa
{
    funcao inicio()
    {
        escreva("Olá Mundo!")
    }
}
```

A screenshot of a code editor window titled 'lista2Ex1.c'. The code is written in C and includes headers for stdio and locale. The main function sets the locale to Portuguese and prints 'Olá Mundo!'.

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      printf("Olá Mundo!");
8      return 0;
9  }
```

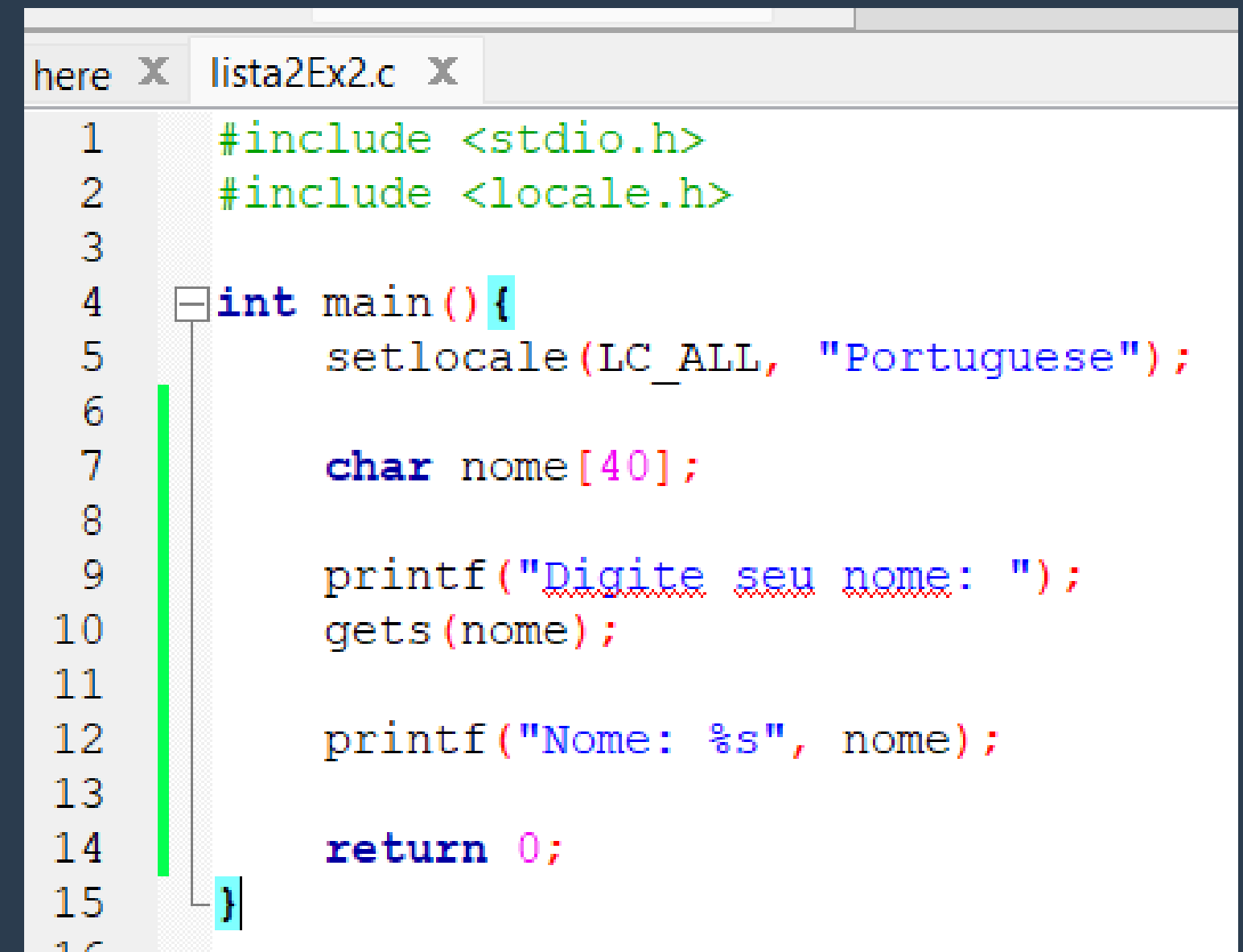
## Exercícios de Revisão:

2) Faça um programa que solicite ao usuário para digitar seu nome. Logo após, exibir o nome digitado.

```
programa
{
    funcao inicio()
    {
        cadeia nome

        escreva("Escreva seu nome ")
        leia(nome)

        escreva("O nome digitado foi ", nome)
    }
}
```



The screenshot shows a code editor window with two tabs: 'here' and 'lista2Ex2.c'. The code in 'lista2Ex2.c' is as follows:

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      char nome[40];
8
9      printf("Digite seu nome: ");
10     gets(nome);
11
12     printf("Nome: %s", nome);
13
14     return 0;
15 }
```

## Exercícios de Revisão:

3) Faça um programa que solicite ao usuário para digitar um número. Logo após, calcular o dobro deste número, depois elevar o valor calculado ao quadrado e exibir:

programa

```
{
    funcao inicio()
    {
        inteiro n, dobro, quadrado

        escreva("Escreva um número ")
        leia(n)
        dobro = n+n
        quadrado = dobro*dobro

        escreva("O quadrado do número é ", quadrado)
    }
}
```

```
[*] Untitled1
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int n=0, dobro=0, quadrado=0;
8
9      printf("Digite um número: ");
10     scanf("%d", &n);
11
12     dobro = n+n;
13     printf("\nDobro = %d", dobro);
14
15     quadrado = dobro * dobro;
16     printf("\nQuadrado = %d", quadrado);
17
18     return 0;
19 }
```

## Exercícios de Revisão:

4) Faça um programa que solicite ao usuário para digitar dois valores, neste caso o primeiro valor é o lado X de um retângulo e o segundo valor o lado Y. Logo após, calcular a área deste retângulo e exibir:

```
Programa {  
    funcao inicio()    {  
        inteiro x, y, area  
  
        escreva("Escreva o lado X do terreno em metros: ")  
        leia(x)  
  
        escreva("\nEscreva o lado Y do terreno em metros: ")  
        leia(y)  
  
        area = x*y  
        escreva("\nA área do terreno é ", area, "m²")  
    }  
}
```

```
1  #include <stdio.h>  
2  #include <locale.h>  
3  
4  int main(){  
5      setlocale(LC_ALL, "Portuguese");  
6  
7      int x=0, y=0, area=0;  
8  
9      printf("Escreva o lado X do terreno em metros: ");  
10     scanf("%d", &x);  
11  
12     printf("Escreva o lado Y do terreno em metros: ");  
13     scanf("%d", &y);  
14  
15     area = x * y;  
16     printf("\nÁrea do terreno = %d m^2", area);  
17  
18     return 0;  
19 }
```

## Exercícios de Revisão:

- 5) Faça um programa que solicite ao usuário para digitar três valores. Logo após, calcular e exibir:
- a) A soma entre os números
  - b) A subtração entre os números
  - c) A multiplicação entre os números
  - d) A divisão conforme fórmula  $(a+b)/c$

# Exercícios de Revisão: Resolução exercício 5

```
programa
{
    funcao inicio()
    {
        real n1, n2, n3, soma, sub, mult, div

        escreva("Digite o primeiro número ")
        leia(n1)
        escreva("Digite o segundo número ")
        leia(n2)
        escreva("Digite o terceiro número ")
        leia(n3)

        soma = n1+n2+n3
        sub = n1-n2-n3
        mult = n1*n2*n3
        div = (n1+n2)/n3

        escreva("\nA) SOMA = ", soma)
        escreva("\nB) SUBTRAÇÃO = ", sub)
        escreva("\nC) MULTIPLICAÇÃO = ", mult)
        escreva("\nD) DIVISÃO = ", div)

    }
}
```

```
[*] lista2Ex5.cpp
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      float n1, n2, n3, soma, sub, mult, div;
8
9      printf("Digite o primeiro número ");
10     scanf("%f", &n1);
11
12     printf("Digite o segundo número ");
13     scanf("%f", &n2);
14
15     printf("Digite o terceiro número ");
16     scanf("%f", &n3);
17
18     soma = n1+n2+n3;
19     sub = n1-n2-n3;
20     mult = n1*n2*n3;
21     div = (n1+n2)/n3;
22
23     printf("\nA) SOMA = %.2f", soma);
24     printf("\nB) SUBTRAÇÃO = %.2f", sub);
25     printf("\nC) MULTIPLICAÇÃO = %.2f", mult);
26     printf("\nD) DIVISÃO = %.2f", div);
27
28     return 0;
29 }
```

## Exercícios de Revisão:

- 6) Faça um programa que solicite ao usuário para digitar um valor. Logo após, utilize a biblioteca Matemática para calcular e exibir:
- a) O dobro do número
  - b) O número elevado ao cubo
  - c) O número elevado ao quadrado
  - d) A raiz cúbica do número

## Exercícios de Revisão:

```
programa
{
    inclua biblioteca Matematica --> mat
    funcao inicio()
    {
        real n1, quadrado, cubo, raizCubica

        escreva("Digite um número: ")
        leia(n1)

        cubo = mat.potencia(n1, 3.0) // Calcula o valor elevado ao cubo
        quadrado = mat.potencia(n1, 2.0) // Calcula o valor elevado ao quadrado
        raizCubica = mat.raiz(n1, 3.0) // Calcula a raiz cúbica do valor

        escreva("\nA) Valor elevado ao cubo = ", cubo)
        escreva("\nB) Valor elevado ao quadrado = ", quadrado)
        escreva("\nC) Raiz Cúbica = ", raizCubica)
    }
}
```

5) Faça um programa que solicite ao usuário para digitar um valor. Logo após, utilize a biblioteca Matemática para calcular e exibir:

- a) O dobro do número
- b) O número elevado ao cubo
- c) O número elevado ao quadrado
- d) A raiz cúbica do número



## Exercícios de Revisão:

6) Faça um programa que solicite ao usuário para digitar dois valores inteiros e os armazene em duas variáveis, exibe esses dois da seguinte forma A = número 1, B = número 2. Logo após, o programa deve trocar os valores contidos nas variáveis entre si e os exibe ao usuário da mesma forma anterior.

## Exercícios de Revisão:

```
Programa {  
    funcao inicio() {  
        inteiro numero1, numero2, temp  
  
        escreva("Digite o número 1: ")  
        leia(numero1)  
        escreva("Digite o número 2: ")  
        leia(numero2)  
  
        //valores antes da troca  
        escreva("\nValores iniciais, Número 1 = ", numero1, " Número 2 = ", numero2)  
  
        //trocando os valores  
        temp = numero1  
        numero1 = numero2  
        numero2 = temp  
        //valores após a troca  
        escreva("\nValores iniciais, Número 1 = ", numero1, " Número 2 = ", numero2)  
    }  
}
```

6) Faça um programa que solicite ao usuário para digitar dois valores inteiros e os armazene em duas variáveis, exibe esses dois da seguinte forma A = número 1, B = número 2. Logo após, o programa deve trocar os valores contidos nas variáveis entre si e os exibe ao usuário da mesma forma anterior.

## Exercícios de Revisão:

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int n1, n2, temp;
8
9      printf("Digite o número 1: ");
10     scanf("%d", &n1);
11
12     printf("Digite o número 2: ");
13     scanf("%d", &n2);
14
15     printf("\nValores iniciais, Número 1 = %d - Número 2 = %d", n1, n2);
16
17     temp = n1;
18     n1 = n2;
19     n2 = temp;
20
21     printf("\nValores após troca, Número 1 = %d - Número 2 = %d", n1, n2);
22     return 0;
23 }
```

6) Faça um programa que solicite ao usuário para digitar dois valores inteiros e os armazene em duas variáveis, exibe esses dois da seguinte forma A = número 1, B = número 2. Logo após, o programa deve trocar os valores contidos nas variáveis entre si e os exibe ao usuário da mesma forma anterior.

Na unidade vimos as **estruturas básicas sequenciais** onde, todos os comandos que são executados seguem uma sequência **linear**, de cima para baixo. Portanto, os comandos são executados na mesma ordem em que foram escritos.

Nos próximos slides iremos trabalhar com **estruturas condicionais**, também chamadas de estruturas de seleção, decisão, ou ainda de desvio condicional.

## ***Tomada de decisões: operadores relacionais e de igualdade***

As instruções executáveis em C tanto realizam ações (como cálculos ou entrada/saída de dados) quanto tomam **decisões** (em breve, veremos alguns exemplos disso). Por exemplo, em um programa, poderíamos tomar a decisão de determinar se a nota de uma pessoa em uma prova seria maior ou igual a 70, e, se fosse, imprimir a mensagem 'Parabéns! Você foi aprovado'.

Essa parte introduz uma versão simples da **estrutura *if*** (ou **instrução *if***), a qual permite que um programa tome decisões com base na veracidade ou falsidade de uma **condição**.

Se a condição é satisfeita (ou seja, se ela for **true** — verdadeira), a instrução no corpo da estrutura *if* é executada. Se a condição não for satisfeita (ou seja, se for **false** — falsa), a instrução do corpo não é executada.

Não importa se o corpo da estrutura é executado ou não; depois que a estrutura *if* termina, a execução prossegue com a próxima instrução após a estrutura *if*.

## Tomada de decisões: operadores relacionais e de igualdade

As condições em estruturas *if* podem ser definidas com o uso dos **operadores de igualdade** e dos **operadores relacionais**, resumidos.

Operador de igualdade ou relacional em C	Exemplo de condição em C	Significado da condição em C
<i>Operadores de igualdade</i>		
<code>==</code>	<code>x == y</code>	x é igual a y
<code>!=</code>	<code>x != y</code>	x não é igual a y
<i>Operadores relacionais</i>		
<code>&gt;</code>	<code>x &gt; y</code>	x é maior que y
<code>&lt;</code>	<code>x &lt; y</code>	x é menor que y
<code>&gt;=</code>	<code>x &gt;= y</code>	x é maior ou igual a y
<code>&lt;=</code>	<code>x &lt;= y</code>	x é menor ou igual a y

■ Operadores de igualdade e relacionais.

## Estrutura de Seleção ou decisão

Uma estrutura de seleção, também chamada de estrutura condicional ou de decisão, é usada para decidir se um conjunto de instruções deve, ou não, ser realizado. Neste tipo de estrutura, pode ocorrer de parte do código de um programa ser executado ou não.

Uma condição simples é formulada com o uso de operadores relacionais

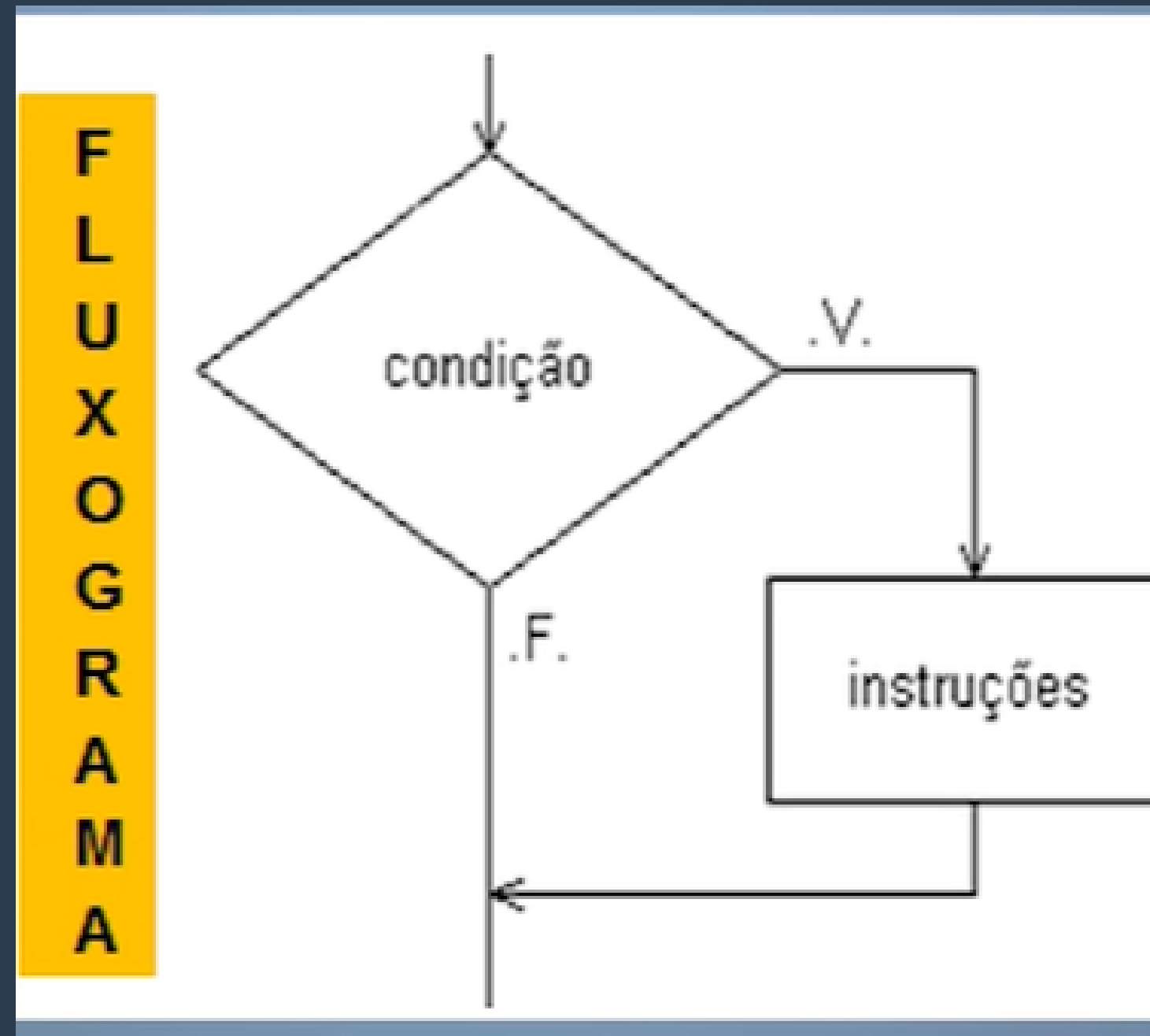
Os símbolos dos **OPERADORES RELACIONAIS** são:

- igual ==
- diferente <> ou !=
- Maior >
- Maior ou igual >=
- Menor <
- Menor ou igual <=

Uma **Condição composta** envolve diversas "condições" conectadas por **OPERADORES LÓGICOS**. Os **OPERADORES LÓGICOS** são: **E, OU, NÃO**.

## *Estrutura de Seleção ou decisão*

### *Desvio condicional simples*



Um desvio condicional é usado para decidir se um conjunto de instruções deve, ou não, ser realizado.



# ***Estrutura de Seleção***

## ***Estrutura de Seleção Simples***

***“SE-ENTÃO” na “linguagem PORTUGOL”***

se (condição verdadeira) então

{

bloco de instruções

}

***Em Portugol Studio***

se (condição verdadeira)

{

bloco de instruções

}

## *Estrutura de Seleção ou decisão*

### *ESTRUTURA DE SELEÇÃO COMPOSTA*

#### *“SE-ENTÃO-SENÃO” na “linguagem PORTUGOL”*

```
Se (condição verdadeira) então
{
    bloco de instruções
}
Senão
{
    bloco de instruções
}
```

Na **seleção composta**, uma condição é avaliada e:

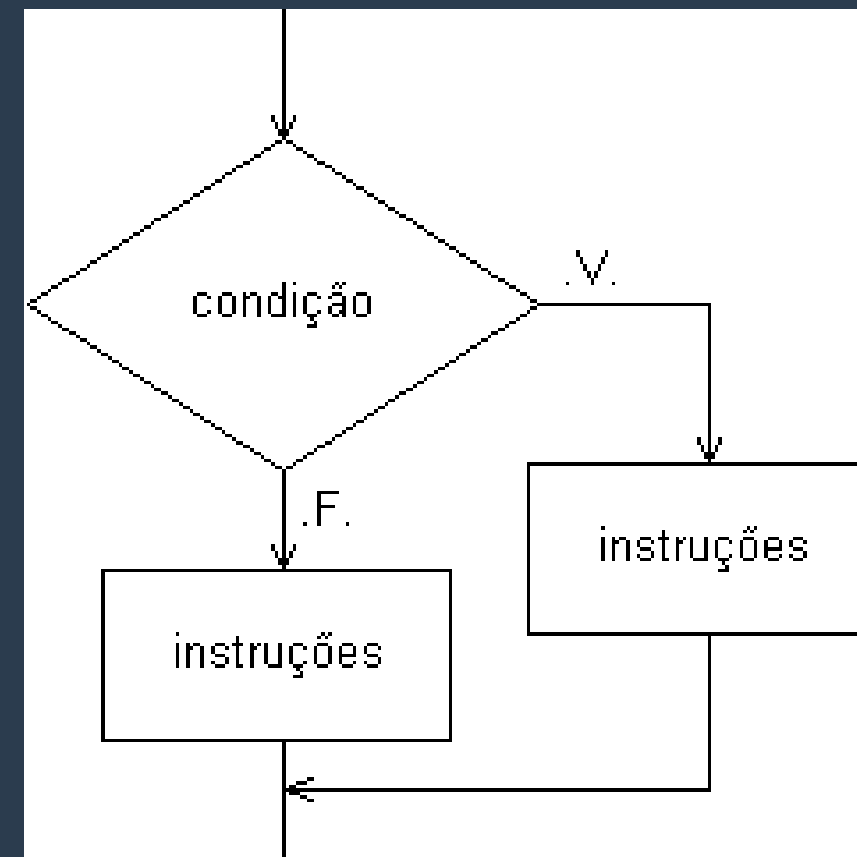
- SE o resultado for verdadeiro,
- ENTÃO um **bloco** de instruções “**A**” é executado.
- SENÃO outro bloco de instruções “**B**” será executado

## *Estrutura de Seleção ou decisão*

### *ESTRUTURA DE SELEÇÃO COMPOSTA*

*“SE-ENTÃO-SENÃO” na “linguagem PORTUGOL”*

```
Se (condição verdadeira) então
{
    bloco de instruções
}
Senão
{
    bloco de instruções
}
```



### Seleção Composta

**Exemplo:** Se a média das notas do aluno for maior ou igual a 6, imprimir “Aprovado”. Caso contrário, imprimir “Reprovado”

#### SINTAXE PORTUGOL

```
se ( media >= 6 ) {  
    escreva( "Aprovado" )  
}  
senao {  
    escreva( "Reprovado" )  
}
```

## Estrutura de Seleção ou decisão

```
1 programa
2 {
3     funcao inicio()
4     {
5         real media = 9
6
7         se(media >= 6){
8             escreva("Olá você foi APROVADO com média: ", media)
9         }
10        senao{
11            escreva("Olá você foi REPROVADO com média: ", media)
12        }
13    }
14 }
15
```

## Estrutura de Seleção ou decisão

```
real nota1, nota2, media

escreva("Digite a primeira nota : ")
leia (nota1)
escreva("Digite a segunda nota : ")
leia (nota2)

media = (nota1 + nota2)/ 2

escreva ("Média do Aluno = ", media)

se (media>=6) {
    escreva(" - APROVADO")
}
senao {
    escreva(" - REPROVADO ")
}
```

## Estrutura de Seleção ou decisão

```
1  /*Programa que mostra na tela se o aluno foi aprovado ou reprovado
2  o usuário deve digitar duas notas para o aluno*/
3  programa
4  {
5      funcao inicio()
6      {
7          real nota1, nota2, media
8          escreva("Digite a primeira nota do aluno: ")
9          leia(nota1)
10
11         escreva("Digite a segunda nota do aluno: ")
12         leia(nota2)
13
14         media = (nota1+nota2)/2
15
16         escreva("Média do Aluno: ", media)
17
18         se(media >= 6){
19             escreva(" - APROVADO \nPARABÉNS")
20         }
21         senao{
22             escreva(" - REPROVADO")
23         }
24     }
25 }
```

## EXERCÍCIO 1

1 - Escreva um programa para calcular o peso ideal de uma pessoa, utilizando as seguintes fórmulas:

homens:  $(72.7 * \text{altura}) - 58$

mulheres:  $(62.1 * \text{altura}) - 44.7$

A saída deve apresentar o resultado do cálculo mostrando apenas o valor para homem caso o usuário digite inicialmente M (sexo masculino), ou para mulher caso digite inicialmente F (sexo feminino). Além disso o usuário deve digitar sua altura.



## EXERCÍCIO 1

### Resolução

```
real altura, pesoideal
character sexo

escreva("Digite o sexo [M ou F] : ")
leia (sexo)
escreva("Digite a altura :")
leia (altura)

se (sexo == 'M') {
    pesoideal = (72.7 * altura) - 58
}
senao {
    pesoideal = (62.1 * altura) - 44.7
}

escreva ("Seu Peso Ideal é =", pesoideal)
```

# Estrutura de Seleção ou decisão

```
1 //comentário, outro exemplo do programa exercício 1.
2 programa
3 {
4     funcao inicio()
5     {
6         caracter sexo
7         real altura, h, m
8
9         escreva("Digite M para Masculino ou F para Feminino: ")
10        leia(sexo)
11
12        escreva("Digite sua altura: ")
13        leia(altura)
14
15        se(sexo == 'M' ou sexo == 'm'){
16            h = (72.7 * altura) - 58
17            escreva("O seu peso ideal é: ", h)
18        }
19        senao se(sexo == 'F' ou sexo == 'f'){
20            m = (62.1 * altura) - 44.7
21            escreva("O seu peso ideal é: ", m)
22        }
23        senao{
24            escreva("Foi digitado uma letra inválida")
25        }
26    }
27 }
```

## Outro Exemplo

### Resolução 2

```
programa
{
    funcao inicio()    {
        real altura = 0.0, homem = 0.0, mulher = 0.0
        caracter tipoSexo
        escreva("Digite M para Masculino ou F para Feminino: ")
        leia(tipoSexo)

        se(tipoSexo == 'M' ou tipoSexo == 'm')    {
            escreva("\nDigite sua altura em metros exatos: ")
            leia(altura)

            homem = (72.7*altura)-58
            escreva("\nPeso ideal para homem: ")
            escreva(homem)
        }
        senao se(tipoSexo == 'F' ou tipoSexo == 'f')    {
            escreva("\nDigite sua altura em metros exatos: ")
            leia(altura)
            mulher = (62.1*altura)-44.7
            escreva("\nPeso ideal para mulher: ")
            escreva(mulher)
        }
        senao    {
            escreva("\nVocê não digitou um tipo de sexo válido\n")
        }
    }
}
```

## EXERCÍCIO 2

2 - Escreva um programa para calcular se um aluno pode ou não ser aprovado.

Para um aluno do curso de Inglês ser aprovado o aluno deve ter média aritmética de duas provas **maior ou igual** a 6 e não pode ter mais do que 12 faltas.

Exiba somente o texto “Aprovado” ou “Reprovado”.

# EXERCÍCIO 2

## Resolução

```
real nota1, nota2, media
inteiro faltas

escreva("Digite a primeira nota : ")
leia (nota1)
escreva("Digite a segunda nota : ")
leia (nota2)
escreva("Digite a quantidade de faltas : ")
leia (faltas)

media = (nota1 + nota2)/ 2

escreva ("Média do Aluno = ", media)

se (media>=6 e faltas <=12) {
    escreva(" - APROVADO")
}
senao {
    escreva(" - REPROVADO ")
}
```

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      float nota1=0.0, nota2=0.0, media=0.0;
8      int faltas=0;
9
10     printf("Digite a primeira nota: ");
11     scanf("%f", &nota1);
12
13     printf("Digite a segunda nota: ");
14     scanf("%f", &nota2);
15
16     printf("Digite a quantidade de faltas: ");
17     scanf("%d", &faltas);
18
19
20     media = (nota1+nota2)/2.0;
21
22     printf("Média do Aluno = %.2f", media);
23
24     if(media >= 6 & faltas <= 12){
25         printf(" - Aprovado!");
26     }
27     else
28         printf(" - Reprovado!");
29
30     return 0;
31 }
```

E:\Aulas\aulas2023\belasArt × + ▾

```
Digite a primeira nota: 9,5
Digite a segunda nota: 5,0
Digite a quantidade de faltas: 1
Média do Aluno = 7,25 - Aprovado!
```

## EXERCÍCIO 2

Outra maneira  
de resolver

```
1 programa
2 {
3     funcao inicio()
4     {
5         real nota1, nota2, media
6         inteiro faltas
7
8         escreva("Digite o número de faltas do aluno: ")
9         leia(faltas)
10
11        escreva("Digite a nota 1 do aluno: ")
12        leia(nota1)
13
14        escreva("Digite a nota 2 do aluno: ")
15        leia(nota2)
16
17        media = (nota1 + nota2)/2
18
19        se(media >= 6 e faltas <= 12){
20            escreva("Aprovado")
21        }
22        senao{
23            escreva("Reprovado")
24        }
25    }
26 }
```

## EXERCÍCIO 3

3 - Escreva um programa que solicita ao usuário para digitar seu nome, após apresenta o nome, além disso, o programa solicita ao usuário para digitar dois valores inteiros **a** e **b** e compara se a variável **a** é diferente da variável **b**. Caso *Verdadeiro*, exibe um texto “A é diferente de B” e a variável **a** é modificada recebendo o valor de **a** mais **b**. Caso *Falso*, exibe o texto “A é igual a B”.

## EXERCÍCIO 3 - Resolução

```
1 //Outro exemplo de resolução para o exercício 3
2 programa
3 {
4     funcao inicio()
5     {
6         cadeia nome
7         inteiro a=0, b=0
8
9         escreva("Digite seu nome: ")
10        leia(nome)
11        limpa()
12        escreva("Nome do aluno: ", nome)
13
14        escreva("\n\nDigite dois números: ")
15        leia(a)
16        leia(b)
17
18        se(a!=b){
19            escreva("A é diferente de B")
20            a = a+b
21        }
22        senao{
23            escreva("\nA é igual a B")
24        }
25    }
26 }
```

```
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main() {
5     setlocale(LC_ALL, "Portuguese");
6
7     char nome[50];
8     int a=0, b=0;
9
10    printf("Digite seu nome: ");
11    gets(nome);
12
13    printf("Digite dois números:\n");
14    scanf("%d", &a);
15    scanf("%d", &b);
16
17    if(a != b) {
18        printf("A é diferente de B");
19
20        a = a+b;
21    }
22    else {
23        printf("A é igual a B");
24    }
25
26    return 0;
27 }
```

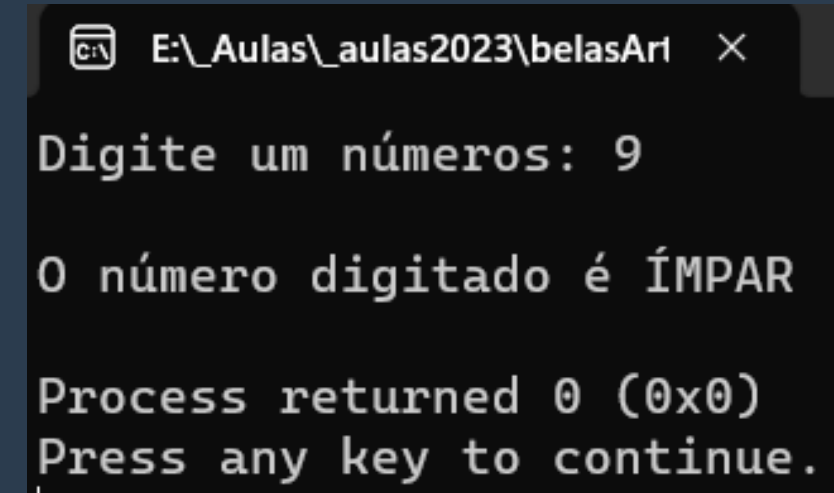
```
E:\Aulas\aulas2023\belasArt x +
Digite seu nome: FABIO F ASSIS
Digite dois números:
2
2
A é igual a B
```

```
E:\Aulas\aulas2023\belasArt x +
Digite seu nome: FABIO F ASSIS
Digite dois números:
5
3
A é diferente de B
Process returned 0 (0x0)   exe
Press any key to continue.
```



## Estrutura de Seleção ou decisão

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      int n=0;
8
9      printf("Digite um números: ");
10     scanf("%d", &n);
11
12     if(n % 2 == 0)
13         printf("\nO número digitado é PAR\n");
14
15     else
16         printf("\nO número digitado é ÍMPAR\n");
17
18     return 0;
19 }
```

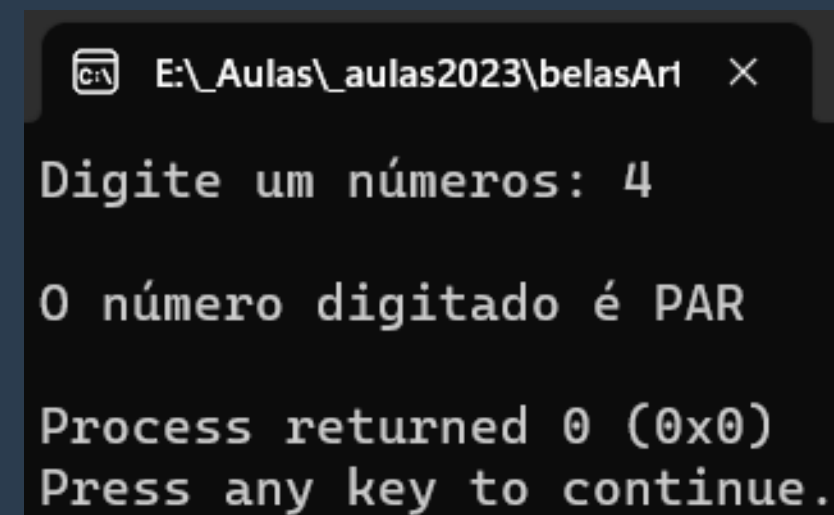


E:\Aulas\aulas2023\belasArt ×

Digite um números: 9

O número digitado é ÍMPAR

Process returned 0 (0x0)  
Press any key to continue.



E:\Aulas\aulas2023\belasArt ×

Digite um números: 4

O número digitado é PAR

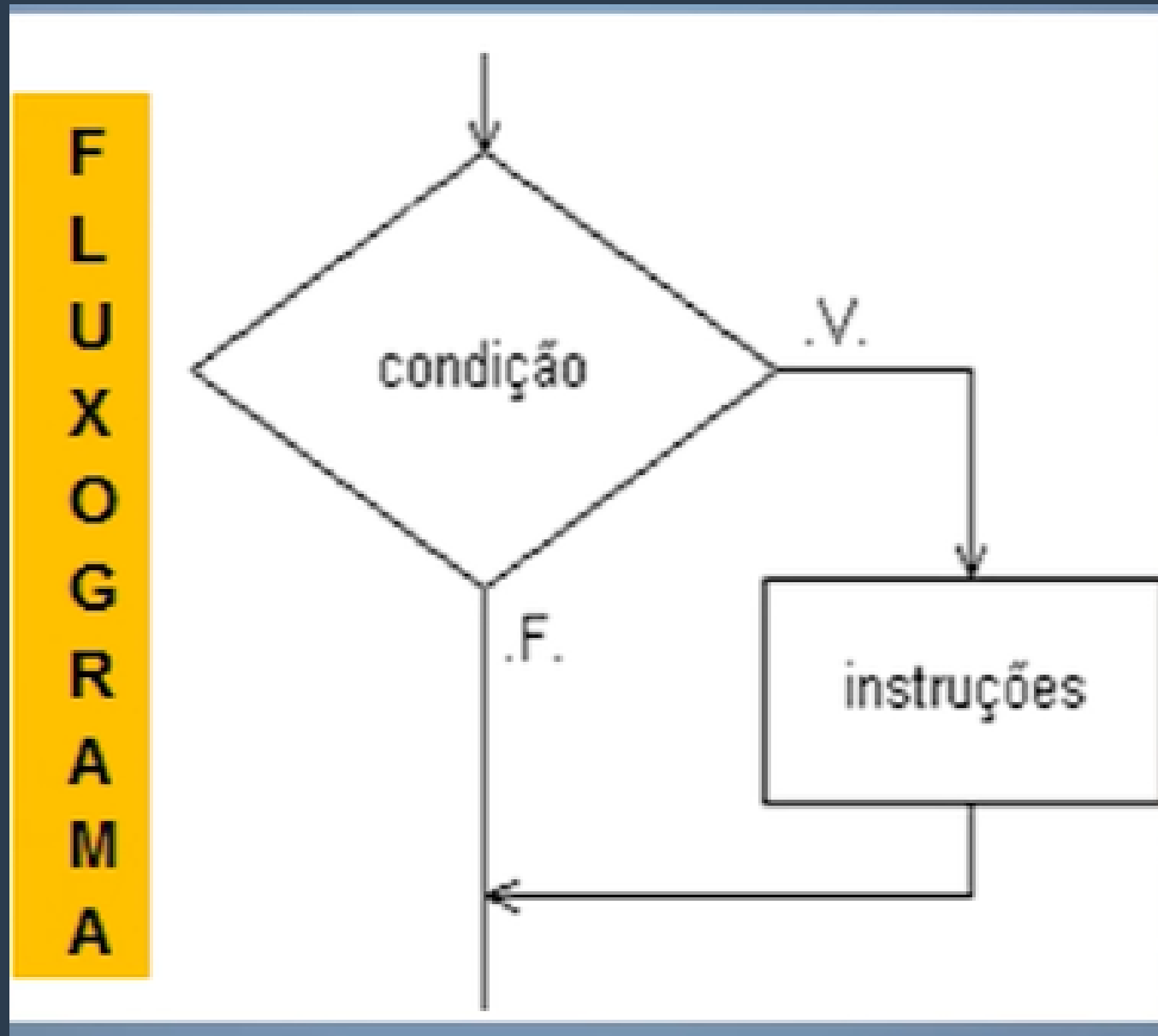
Process returned 0 (0x0)  
Press any key to continue.

Este exemplo pede ao usuário que digite um número. Depois verifica se o número digitado é Par ou se é Ímpar e imprime esta informação.

# Estrutura de Seleção ou decisão

## DESVIO CONDICIONAL SIMPLES

Um desvio condicional é usado para decidir se um conjunto de instruções deve, ou não, ser realizado.



```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6      int a = 0;
7      if(a>0)
8          printf("A é maior ou igual a zero");
9
10     printf("\nContinuação do programa!\n");
11     return 0;
12 }
13
14
15
```

C:\Users\ffass\Desktop\Unittl ×

Continuação do programa!

Process returned 0 (0x0)  
Press any key to continue.

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6      int a = 1;
7      if(a>0)
8          printf("A é maior que zero");
9
10     printf("\nContinuação do programa!\n");
11     return 0;
12 }
13
14
15
```

C:\Users\ffass\Desktop\Unittl ×

A é maior que zero  
Continuação do programa!

Process returned 0 (0x0)  
Press any key to continue.

## Estrutura de Seleção ou decisão

### ESTRUTURA DE SELEÇÃO COMPOSTA

#### “SE-ENTÃO-SENÃO” na “linguagem PORTUGOL”

Na **seleção composta**, uma condição é avaliada e:

- **SE** o resultado for verdadeiro,
- **ENTÃO** um **bloco** de instruções “**A**” é executado.
- **SENÃO** outro bloco de instruções “**B**” será executado

Se (condição verdadeira) então

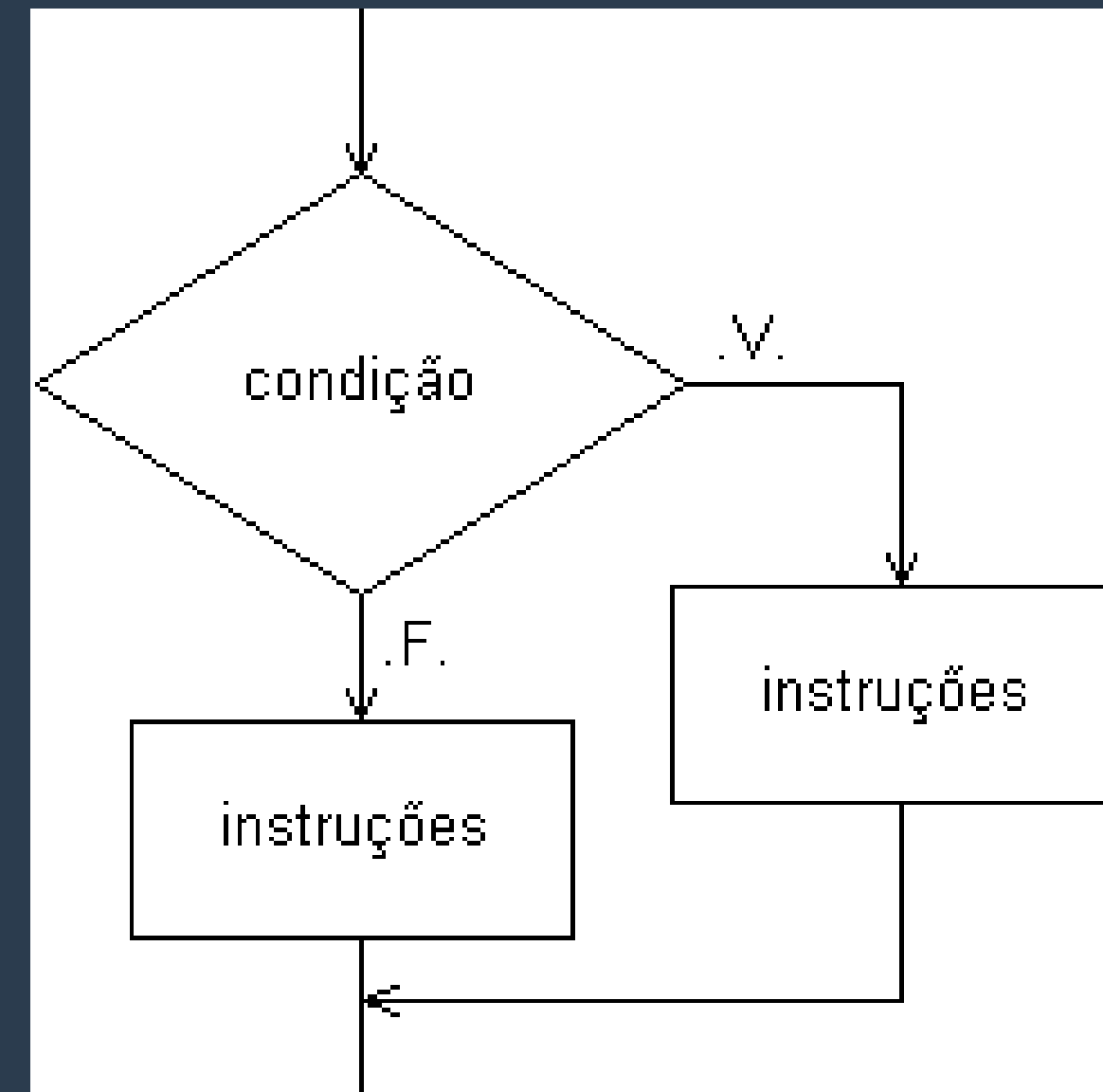
```
{  
    bloco de instruções
```

```
}
```

Senão

```
{  
    bloco de instruções
```

```
}
```



# Estrutura de Seleção ou decisão

## ESTRUTURA DE SELEÇÃO COMPOSTA

### “SE-ENTÃO-SENÃO”

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      int a = 1;
8
9      if(a > 0) {
10         printf("A é maior que zero");
11     }
12
13     else
14         printf("A é igual ou menor que zero");
15
16     printf("\nContinuação do programa!\n");
17
18     return 0;
19 }
```

C:\Users\ffass\Desktop\Un titl ×

A é maior que zero  
Continuação do programa!

Process returned 0 (0x0)  
Press any key to continue.

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main() {
5      setlocale(LC_ALL, "Portuguese");
6
7      int a = 0;
8
9      if(a > 0) {
10         printf("A é maior que zero");
11     }
12
13     else
14         printf("A é igual ou menor que zero");
15
16     printf("\nContinuação do programa!\n");
17
18     return 0;
19 }
```

C:\Users\ffass\Desktop\Un titl ×

A é igual ou menor que zero  
Continuação do programa!

Process returned 0 (0x0)  
Press any key to continue.

# Estrutura de Seleção ou decisão

## Seleção Composta

**Exemplo:** Se a média das notas do aluno for maior ou igual a 6, imprimir “Aprovado”. Caso contrário, imprimir “Reprovado”

### SINTAXE PORTUGOL

```
se ( media >= 6 ) {  
    escreva( "Aprovado" )  
}  
senao {  
    escreva( "Reprovado" )  
}
```



```
1 programa {  
2     funcao inicio() {  
3         inteiro media  
4  
5         escreva("Digite o valor da média: ")  
6         leia(media)  
7  
8         se(media >= 6)  
9             escreva("Aprovado!")  
10  
11        senao  
12            escreva("Reprovado!")  
13    }  
14 }
```

O programa está aguardando a entrada de dados

Digite o valor da média: 7  
Aprovado!  
Programa finalizado. Tempo de execução: 4396 milissegundos



# Estrutura de Seleção ou decisão

Programa que calcula a média entre duas notas reais, no fim imprime a mensagem Aprovado, para o caso onde a média seja maior ou igual a 6, caso contrário imprime reprovado.

```
real nota1, nota2, media

escreva("Digite a primeira nota : ")
leia (nota1)
escreva("Digite a segunda nota : ")
leia (nota2)

media = (nota1 + nota2)/ 2

escreva ("Média do Aluno = ", media)

se (media>=6) {
    escreva(" - APROVADO")
}
senao {
    escreva(" - REPROVADO ")
}
```

```
here X  Untitled1.c X  media1.c X
1      #include <stdio.h>
2      #include <locale.h>
3
4      int main(){
5          setlocale(LC_ALL, "Portuguese");
6
7          float nota1, nota2, media;
8
9          printf("Digite a primeira nota: ");
10         scanf("%f", &nota1);
11
12         printf("Digite a segunda nota: ");
13         scanf("%f", &nota2);
14
15         media = (nota1+nota2)/2;
16
17         printf("Média do Aluno = %.1f", media);
18
19         if(media >= 6){
20             printf(" - Aprovado!");
21         }
22         else
23             printf(" - Reprovado!");
24
25         return 0;
26     }
```

## Estrutura de Seleção ou decisão

```
//programa deve receber duas notas do tipo inteiro,  
//imprimir a média das notas e qual a maior nota  
#include <stdio.h>  
int main() {  
    int n1, n2;  
    float media;  
  
    printf("Digite duas notas: ");  
    scanf("%d %d", &n1, &n2);  
  
    media = (n1+n2)/2.0;  
    printf("Média = %.1f\n", media);  
  
    if(n1==n2)  
        printf("Notas iguais\n");  
    else if(n1>n2)  
        printf("N1 maior = %d", n1);  
    else  
        printf("N2 maior = %d", n2);  
  
    return 0;  
}
```

## Estrutura de Seleção ou decisão

```
//programa deve receber três notas do tipo inteiro,  
//imprimir a média das notas e qual a maior nota  
#include <stdio.h>  
int main(){  
    int n1, n2, n3;  
    float media;  
  
    printf("Digite três notas: ");  
    scanf("%d %d %d", &n1, &n2, &n3);  
  
    media = (n1+n2+n3)/3.0;  
    printf("Média = %.1f\n", media);  
  
    if(n1>n2)  
        if(n1>n3)  
            printf("N1 maior = %d", n1);  
        else  
            printf("N3 maior = %d", n3);  
    else if(n2>n3)  
        printf("N2 maior = %d", n2);  
    else  
        printf("N3 maior = %d", n3);  
  
    return 0;  
}
```



## EXERCÍCIO 1

1 - Escreva um programa para calcular o peso ideal de uma pessoa, utilizando as seguintes fórmulas:

homens:  $(72.7 * \text{altura}) - 58$

mulheres:  $(62.1 * \text{altura}) - 44.7$

A saída deve apresentar o resultado do cálculo mostrando apenas o valor para homem caso o usuário digite inicialmente M (sexo masculino), ou para mulher caso digite inicialmente F (sexo feminino). Além disso o usuário deve digitar sua altura.

## EXERCÍCIO 1

### Resolução

```
real altura, pesoideal
caracter sexo

escreva("Digite o sexo [M ou F] : ")
leia (sexo)
escreva("Digite a altura :")
leia (altura)

se (sexo == 'M') {
    pesoideal = (72.7 * altura) - 58
}
senao {
    pesoideal = (62.1 * altura) - 44.7
}

escreva ("Seu Peso Ideal é =", pesoideal)
```

## EXERCÍCIO 2

2 - Escreva um programa para calcular se um aluno pode ou não ser aprovado.

Para um aluno do curso de Inglês ser aprovado o aluno deve ter média aritmética de duas provas **maior ou igual** a 6 e não pode ter mais do que 12 faltas.

Exiba somente o texto “Aprovado” ou “Reprovado”.

## EXERCÍCIO 2

### Resolução

```
real nota1, nota2, media
inteiro faltas

escreva("Digite a primeira nota : ")
leia (nota1)
escreva("Digite a segunda nota : ")
leia (nota2)
escreva("Digite a quantidade de faltas : ")
leia (faltas)

media = (nota1 + nota2)/ 2

escreva ("Média do Aluno = ", media)

se (media>=6 e faltas <=12) {
    escreva(" - APROVADO")
}
senao {
    escreva(" - REPROVADO ")
}
```

## EXERCÍCIO 3

3 - Este exemplo pede ao usuário que informe um número inteiro. Logo após, exibe uma mensagem indicando se o número informado é positivo, negativo ou igual a zero.

## EXERCÍCIO 3

### Resolução

```
programa
{
    funcao inicio()
    {
        inteiro numero

        escreva("Digite um número inteiro: ")
        leia(numero)

        se(numero > 0) // Verifica se o número é positivo
        {
            escreva("O número é positivo")
        }
        senao se(numero < 0) // Verifica se o número é negativo
        {
            escreva("O número é negativo")
        }
        senao // Se não é positivo nem negativo, só pode ser igual a zero
        {
            escreva("O número é igual zero")
        }

        escreva("\n")
    }
}
```

## EXERCÍCIO 4

4 - Este exemplo pede ao usuário que informe três notas. Logo após, calcula e exibe a média final destas notas. Por último, verifica se alguma das notas parciais é menor que a média final e a exibe (caso exista).

## EXERCÍCIO 4

### Resolução

```
programa {  
  inclui biblioteca Matematica --> mat  
  
  funcao inicio () {  
    real m1, m2, m3, media  
  
    escreva ("Informe a nota 1: " )  
    leia (m1)  
    escreva( "Informe a nota 2: " )  
    leia (m2)  
    escreva ("Informe a nota 3: " )  
    leia (m3)  
  
    media = (m1 + m2 + m3) / 3  
    limpa()  
    escreva ("A média final é: ", mat.arredondar(media, 2), "\n\n")  
  
    se (m1 < media) {  
      escreva ("A média 1 é menor que a média final\n")  
    }  
    se (m2 < media) {  
      escreva ("A média 2 é menor que a média final\n")  
    }  
    se (m3 < media) {  
      escreva ("A média 3 é menor que a média final\n")  
    }  
  }  
}
```





## EXERCÍCIO 5

5 - Este exemplo pede ao usuário que informa a sua idade. Logo após exibe uma mensagem informando se ele é maior ou menor de idade.

## EXERCÍCIO 5

### Resolução

```
programa
{
    funcao inicio ()
    {
        inteiro menor, idade

        escreva("Informe sua idade: ")
        leia(idade)

        se (idade < 18)
        {
            escreva("Você é menor de idade")
        }
        senao
        {
            escreva("Você é maior de idade")
        }

        escreva("\n")
    }
}
```

## Exercícios:

### ***LISTA DE EXERCÍCIOS – NÃO VALE NOTA***

Faça a lista de programa em C.

- 1) Faça um programa que exibi a mensagem "Olá mundo!".
- 2) Faça um programa que solicite ao usuário para digitar seu nome. Logo após, exibir o nome digitado.
- 3) Faça um programa que solicite ao usuário para digitar um valor. Logo após, calcular o dobro deste número e exibir:
- 4) Faça um programa que solicite ao usuário para digitar dois valores, neste caso o primeiro valor é o lado X de um retângulo e o segundo valor o lado Y. Logo após, calcular a área deste retângulo e exibir:

## Exercícios:

- 5) Faça um programa que solicite ao usuário para digitar três valores. Logo após, calcular e exibir:
  - a) A soma entre os números
  - b) A subtração entre os números
  - c) A multiplicação entre os números
  - d) A divisão entre os números
  
- 6) Faça um programa que solicite ao usuário para digitar um valor. Logo após, utilize a biblioteca Matemática para calcular e exibir:
  - a) O número elevado ao cubo
  - b) O número elevado ao quadrado
  - b) A raiz cúbica do número
  
- 7) Faça um programa que solicite ao usuário para digitar dois valores inteiros e os armazene em duas variáveis, exibe esses dois da seguinte forma A = número 1, B = número 2. Logo após, o programa deve trocar os valores contidos nas variáveis entre si e os exibe ao usuário da mesma forma anterior.

## Estruturas de Repetição:

A estrutura de repetição é utilizada quando precisamos repetir partes do programa. Com a vantagem de minimizar o número de linhas de código, deixando-o mais legível.

No Portugol Studio, existem três estruturas de repetição, são elas:

Enquanto (while)

Faça enquanto (do while)

Para (for)

## Estruturas de Repetição:

### ***ENQUANTO***

O bloco do comando *enquanto* é executado, até que a condição seja falsa.

É necessário inicializar a variável de **controle** antes da execução o bloco.

É necessário alterar o valor da variável de controle dentro do bloco de repetição, para que em um determinado valor esta faça com que a condição seja falsa para sair do bloco de repetição.

Caso contrário o código poderá ficar em laços infinito.

## EXEMPLO 5

### EXEMPLO DE ESTRUTURA DE REPETIÇÃO ENQUANTO

Programa que executa um laço de repetição que imprime "\*", até que a condição seja falsa.

programa

```
{  
  funcao inicio()  
  {  
    inteiro c // variável de controle  
    c = 1 // iniciando a variável de controle  
    enquanto (c<=3)  
    {  
      escreva("*")  
      c = c + 1;  
    }  
  }  
}
```

Inicialização

Condição

Incremento

Exemplo em C

```
1  #include <stdio.h>  
2  
3  int main(){  
4      int c = 1;  
5  
6      while(c <= 3){  
7          printf("*");  
8          c = c + 1;  
9      }  
10  
11      return 0;  
12 }
```

# EXEMPLO 5 – TESTE DE MESA

## EXEMPLO DE ESTRUTURA DE REPETIÇÃO ENQUANTO

Programa que executa um laço de repetição que imprime “\*”, até que a condição seja falsa.

```
programa
{
  funcao inicio()
  {
    inteiro c // variável de controle
    c = 1 // iniciando a variável de controle
    enquanto (c<=3)
    {
      escreva("*")
      c = c + 1;
    }
  }
}
```

Tabela 1 – Teste de Mesa

Variável	Condição	
c = 1	c <= 3 = true	*
c = c + 1 = 2	c <= 3 = true	*
c = c + 1 = 3	c <= 3 = true	*
c = c + 1 = 4	c <= 3 = false	



## EXEMPLO 5 – TESTE DE MESA

### *EXEMPLO DE ESTRUTURA DE REPETIÇÃO ENQUANTO*

Imprime a tabuada de um determinado número digitado pelo usuário.

```
1 programa {
2     funcao inicio() {
3         inteiro n1, c=0
4         escreva("Digite um número: ")
5         leia(n1)
6
7         enquanto(c <= 10){
8             escreva(n1, " x ", c, " = ", n1*c, "\n")
9             c++
10        }
11    }
12 }
```

```
> _ Console Mensagens
Digite um número: 3
3 x 0 = 0
3 x 1 = 3
3 x 2 = 6
3 x 3 = 9
3 x 4 = 12
3 x 5 = 15
3 x 6 = 18
3 x 7 = 21
3 x 8 = 24
3 x 9 = 27
3 x 10 = 30

Programa finalizado.
```

## Estruturas de Repetição:

### ***FAÇA ENQUANTO***

O bloco do comando para enquanto é executado, até que a condição seja falsa.

*Exemplo:*

*condicao = 4*

*faca {*

*bloco de instruções*

*condicao++*

*} enquanto (condicao < 10)*

Inicialização

Incremento

Condição

## EXEMPLO 6 - Resolução com Faça Enquanto

Programa que solicita ao usuário para digitar um número  $n$ . Logo após, calcula e exibe na tela a tabuada do número. Resolução com estrutura de repetição

Faça enquanto

```
inteiro n, c=0  
escreva("Qual número você deseja gerar a tabuada? ")  
leia(n)  
faca {  
    escreva(n, " x ", c, " = ", c*n, "\n")  
    c = c + 1  
}enquanto(c<=10)
```

Inicialização

Incremento

Condição

# Estruturas de Repetição:

## ***PARA***

O bloco do comando para é executado enquanto uma condição for verdadeira.

Exemplo:

```
para (variável = valorInicial; variável < valorFinal; variável++)
```

```
{
```

bloco de instruções

```
}
```



Inicialização

Condição

Incremento

## EXEMPLO 7

### *EXEMPLO DE ESTRUTURA DE REPETIÇÃO PARA*

Programa que executa um laço de repetição que imprime “\*”, até que a condição seja falsa.

```
1 programa
2 {
3     funcao inicio()
4     {
5         inteiro c
6         para(c=1; c<=3; c++)
7         {
8             escreva("Olá Mundo\n")
9         }
10    }
11 }
```

Diagram illustrating the structure of a 'para' loop with three components:

- Inicialização** (Initialization): `c=1`
- Condição** (Condition): `c<=3`
- Incremento** (Increment): `c++`

## EXEMPLO 5 – TESTE DE MESA

### *EXEMPLO DE ESTRUTURA DE REPETIÇÃO PARA*

Imprime a tabuada de um determinado número digitado pelo usuário.

```
1 programa {
2     funcao inicio() {
3         inteiro n1, c=0
4         escreva("Digite um número: ")
5         leia(n1)
6
7         para(c=0; c <= 10; c++){
8             escreva(n1, " x ", c, " = ", n1*c, "\n")
9         }
10    }
11 }
```

```
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main(){
5     setlocale(LC_ALL, "Portuguese");
6
7     int n1=0, c=0;
8     printf("Digite um número: ");
9     scanf("%d", &n1);
10
11     for(c=0; c <= 10; c++){
12         printf("%d x %d = %d\n", n1, c, n1*c);
13     }
14
15     return 0;
16 }
```

> \_ Console Mensagens

Digite um número: 3

3 x 0 = 0  
3 x 1 = 3  
3 x 2 = 6  
3 x 3 = 9  
3 x 4 = 12  
3 x 5 = 15  
3 x 6 = 18  
3 x 7 = 21  
3 x 8 = 24  
3 x 9 = 27  
3 x 10 = 30

Programa finalizado.

## EXEMPLO 8

### EXEMPLO DE ESTRUTURA DE REPETIÇÃO EM C

```
ex1_For.cpp
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(void)
5  {
6      setlocale(LC_ALL, "Portuguese");
7      int i;
8
9      for(i=0; i<=10; i++)
10     {
11         printf("Valor de i = %d\n", i);
12     }
13
14     return 0;
15 }
```

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(void)
5  {
6      setlocale(LC_ALL, "Portuguese");
7      int i=0;
8
9      while(i<=10)
10     {
11         printf("Valor de i = %d\n", i);
12         i++;
13     }
14
15     return 0;
16 }
```

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(void)
5  {
6      setlocale(LC_ALL, "Portuguese");
7      int i=0;
8
9      do
10     {
11         printf("Valor de i = %d\n", i);
12         i++;
13     }while(i<=10);
14
15     return 0;
16 }
```

Saída na tela

```
Valor de i = 0
Valor de i = 1
Valor de i = 2
Valor de i = 3
Valor de i = 4
Valor de i = 5
Valor de i = 6
Valor de i = 7
Valor de i = 8
Valor de i = 9
Valor de i = 10
```

# Estruturas de Repetição:

## FOR

```
Untitled1  x  [*] ExemploFor6.c  x
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int i=0;
8
9      for(i=0; i < 3; i++){
10         printf("*\n");
11     }
12
13     return 0;
14 }
```

```
</> source code
1  #include <stdio.h>
2
3  int main()
4  {
5      int i = 0;
6
7      for(i=0; i<3; i++)
8      {
9          if(i!=2)
10             printf("*\n");
11          else
12             printf("*");
13      }
14
15     return 0;
16 }
```



# Estruturas de Repetição:

## Exemplo de média de 10 números

```
programa
{
    funcao inicio()
    {
        inteiro contador = 1

        real numero, media, soma = 0.0

        // Laço que verifica se já foram informados 10 valores

        enquanto(contador <= 10)
        {
            limpa()
            escreva("Digite o ", contador, "º número: ")
            leia(numero)

            soma = soma + numero    // A variavel soma é o acumulador deste exemplo
            contador = contador + 1 // Incrementa o contador
        }

        media = soma / 10

        limpa()
        escreva("A média dos números é: ", media, "\n")
    }
}
```

```
Untitled1 × ExemploWhile1.c ×
1 #include <stdio.h>
2 #include <locale.h>
3
4 int main(){
5     setlocale(LC_ALL, "Portuguese");
6
7     int contador=1, numero=0, soma=0;
8     float media = 0.0;
9
10    while(contador <= 10){
11        printf("Digite o %dº número: ", contador);
12        scanf("%d", &numero);
13
14        soma = soma + numero;
15        contador++;
16    }
17
18    media = soma/10.0;
19    printf("A média dos números é: %.2f", media);
20    return 0;
21 }
```

# Estruturas de Repetição:

## Exemplo de média de 10 números

```
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int contador=1, numero=0, soma=0;
8      float media = 0.0;
9
10     for(contador = 1; contador <= 10; contador++){
11         printf("Digite o %dº número: ", contador);
12         scanf("%d", &numero);
13
14         soma = soma + numero;
15     }
16
17     media = soma/10.0;
18     printf("A média dos números é: %.2f", media);
19     return 0;
20 }
```

```
Untitled1  ×  ExemploWhile1.c  ×
1  #include <stdio.h>
2  #include <locale.h>
3
4  int main(){
5      setlocale(LC_ALL, "Portuguese");
6
7      int contador=1, numero=0, soma=0;
8      float media = 0.0;
9
10     while(contador <= 10){
11         printf("Digite o %dº número: ", contador);
12         scanf("%d", &numero);
13
14         soma = soma + numero;
15         contador++;
16     }
17
18     media = soma/10.0;
19     printf("A média dos números é: %.2f", media);
20     return 0;
21 }
```

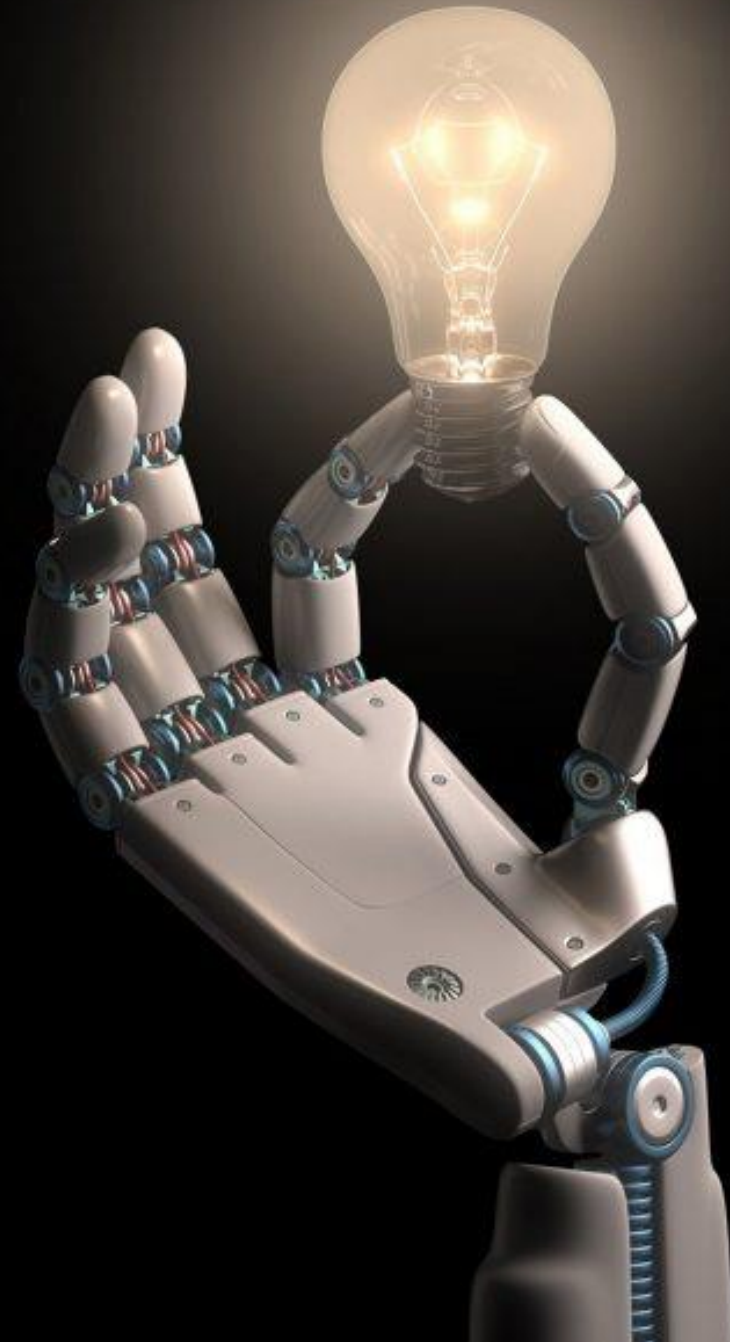
## Resumo

Na unidade III vimos as **estruturas de repetição**, for, while e do while.

Na próxima unidade iremos ver vetor e matriz

# Fim

**A Computação ilumina sua mente**



**Agradeço a atenção!**

fabio.assis@online.uscs.edu.br

Figura 4 – Computação x robótica (Adaptado)  
Fonte: <https://tecnico.ulisboa.pt/pt/tag/robotica/>

# ***Bibliografia***

## Bibliografia Básica

1 - Entendendo Algoritmos: Um Guia Ilustrado Para Programadores e Outros Curiosos. Autor: Aditya Bhargava. Editora: Novatec Editora; 1ª edição (24 abril 2017)

link para download:

<https://br1lib.org/book/17415411/68ed2b>

<https://ler-livros.com/ler-online-ebook-pdf-entendendo-algoritmos-um-guia-ilustrado-para-programadores-e-outros-curiosos-baixar-resumo/>

2 - C - Como Programar [6ª Ed][Deitel & Deitel][2011]. Autor: Paul Deitel, Harvey Deitel. Editora: Pearson; 6ª edição (6 agosto 2015)

3 - Introdução à Programação com Python: Algoritmos e Lógica de Programação Para Iniciantes

Autor: Nilo Ney Coutinho Menezes. Editora: Novatec Editora; 3ª edição (8 janeiro 2019)

## Complementar

1 - CORMEN, T H LEISERSON, C E RIVEST, R L STEIN, C Algoritmos teoria e prática 2ª edição Rio de Janeiro, RJ Campus, 2002

2 - Lógica de programação com Portugol. Autor: Joice Barbosa Mendes, Rafael da Silva Muniz

3 - Algoritmos E Lógica Da Programação

Autor: Marco A. Furlan de Souza (Autor), Marcelo Marques Gomes (Autor), Marcio Vieira Soares (Autor), Ricardo Concilio (Autor)

Editora: Cengage Learning; 3ª edição (10 janeiro 2019)

4 - Wes McKinney

Python Para Análise de Dados Tratamento de Dados com Pandas, NumPy e IPython Edição Português, Novatec Editora, 9 janeiro 2018

5 - Aprenda Lógica de Programação e Algoritmos com Implementações em Portugol, Scratch, C, Java, C# e Python

Autor: Cláudio Luís Vieira Oliveira. Editora: Ciência Moderna (1 janeiro 2016)