



# LABORATÓRIO DE PROGRAMAÇÃO

## Variáveis e Comandos em C

Profº. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

1º. Período

# Laboratório de Programação

## Variáveis e Comandos em C

### Variáveis

- ✓ São o aspecto fundamental de qualquer **linguagem de programação**.
- ✓ É um espaço de memória reservado para armazenar um certo tipo de dado
  - Tendo um nome para referenciar o seu conteúdo.
- ✓ São objetos que podem assumir diferentes valores dentro do mesmo programa.
  - Apenas um valor por vez.

# Laboratório de Programação

## Variáveis e Comandos em C

### Variáveis

- ✓ Cada variável possui um **tipo** e um **identificador (nome)** válido.
- ✓ Toda variável deve ser **declarada** antes de ser utilizada
  - Na declaração, associa-se o tipo e identificador.
- ✓ A forma geral de declarar uma variável é:

*tipo\_de\_dados lista\_de\_variáveis;*

# Laboratório de Programação

## Variáveis e Comandos em C

### Variáveis

✓ Uma variável:

➤ **[tipo]** [identificador] ;

✓ Mais de uma variável do mesmo tipo:

➤ **[tipo]** [identificador1] , [identificador2] ;

✓ Variáveis de diferentes tipos:

➤ **[tipo 1]** [identificador1] , [identificador2] ;

➤ **[tipo 2]** [identificador3] , [identificador4] ;



# Laboratório de Programação

## Variáveis e Comandos em C

### Variáveis – Exemplos

- ✓ `int numerointeiro;` //variável do tipo inteiro
- ✓ `int Numerolnteiro;` //variável do tipo inteiro
- ✓ `float nota , temperatura;` //variáveis do tipo real
- ✓ `char letra;` //variável do tipo caractere
- ✓ `char nome[50];` //variável do tipo string (conjunto de caracteres)

**A linguagem C é case sensitive. A linguagem diferencia letras maiúsculas de letras minúsculas.**

# Laboratório de Programação

## Variáveis e Comandos em C

### Variáveis

- ✓ Onde as variáveis são declaradas?
  - Por enquanto, iremos trabalhar com variáveis locais.
- ✓ São declaradas dentro do bloco da função **main**.
  - Exemplo:

```
#include<stdio.h>
#include<stdlib.h>
int main ( ) {
    Variável → int num;
                  num = 10; ← Atribuição
    printf("O número informado é %d", num);
    system ("pause");
    return (0);
}
```

### Variáveis Constantes

- ✓ **Semelhantes as variáveis, as constantes representam valores de um único tipo.**
- ✓ **Porém como o próprio nome já diz, as constantes armazenam/representam um único valor durante todo o programa.**
  - **Por isso, não existe atribuição de valor para uma constante.**

# Laboratório de Programação

## Variáveis e Comandos em C

### Variáveis Constantes

Constantes →  
Globais

```
#include<stdio.h>
#define TEMPERATURA 4.9
#define NOME "Sergio Vieira"
int main ( )
{
}
}
```

- ✓ Sempre antes da função **main**
- ✓ Sempre usa-se **#define**
- ✓ Usualmente em maiúsculas
- ✓ Qualquer tipo e valor
- ✓ Não tem o **“;”**
- ✓ Não ocupa espaço em memória



# Laboratório de Programação

## Variáveis e Comandos em C

### Variáveis Constantes

Constantes →  
Locais

```
#include<stdio.h>
int main ( )
{
    const TEMPERATURA 4.9;
    const VERDADEIRO -1;
    //const NOME "Sergio Veira"; // Não funciona
    return (0);
}
```

- ✓ Dentro da função **main**
- ✓ Usualmente em maiúsculas
- ✓ Qualquer tipo e valor, exceto *strings*
- ✓ Tem o “**;**”
- ✓ Ocupa espaço em memória

# Laboratório de Programação

## Variáveis e Comandos em C

### Variáveis

#### ✓ Por que declarar variáveis?

- Reunir variáveis em um mesmo lugar, dando a elas nomes significativos, facilita ao leitor entender o que o programa faz.
- Uma seção de declarações de variáveis encoraja o planejamento do programa antes de começá-lo.
- Declarar variáveis ajuda a prevenir erros. Por exemplo, se escrevermos **o** (letra o) em vez de **0** (zero).

```
int b0b0;
```

```
bobo = 5;
```

O compilador acusará o erro.

# Laboratório de Programação

## Variáveis e Comandos em C

### Atribuição

- ✓ O operador de atribuição é o '=' (representado por ← nos algoritmos em português).

- Lê-se “recebe”.
- Não confundir com o sinal de igualdade ('==')

//Exemplos de atribuição de valores

**numero\_inteiro = 10;** //variável numero\_inteiro recebe 10

**nota = 6.6;** //variável nota recebe 6.6

**letra = 'a';** //variável letra recebe a

**sexo = “Masculino”;** //variável sexo recebe Masculino

# Laboratório de Programação

## Variáveis e Comandos em C

### Inicialização

- ✓ Ao mesmo tempo que declaramos uma variável, podemos atribuir a ela valores.
  - Isto se chama de inicialização de uma variável.
- ✓ Quando atribuímos valores a uma variável ao longo de um programa, utilizamos apenas o termo “atribuição” ou “recebe”.

//inicialização das variáveis

**int** **numero\_inteiro** = **10**; //variável numero\_inteiro recebe 10

**float** **nota** = **6.6**; //variável nota recebe 6.6

**char** **letra** = **'a'**; //variável letra recebe a

**char[20]** **sexo** = **“Masculino”**; //variável sexo recebe Masculino

# Laboratório de Programação

## Variáveis e Comandos em C

### Inicialização

✓ Onde as variáveis são inicializadas?

- No momento em que são declaradas, coloca-se o sinal de = (igual) e uma constante após o nome.

✓ A forma geral de uma inicialização é:

**tipo nome\_variável = valor\_constante;**

```
#include<stdio.h>
#include<stdlib.h>
int main ( ) {
    int num=10; ← Inicialização
    printf("O número informado é %d", num);
    system("pause");
    return (0);
}
```

```
#include<stdio.h>
#include<stdlib.h>
int main ( ) {
    int num;
    num = 10; ← Atribuição
    printf("O número informado é %d ", num);
    system("pause");
    return (0);
}
```



# Laboratório de Programação

## Variáveis e Comandos em C

### Comandos de Entrada e Saída

- ✓ Para utilizarmos instruções de entrada e saída, por exemplo, ler e imprimir dados, antes precisamos incluir a biblioteca **stdio.h** em nosso programa.

- **#include<stdio.h>**

- ✓ Ela contém as definições das instruções necessárias.

- **scanf()**

- **printf()**

# Laboratório de Programação

## Variáveis e Comandos em C

### Comando de Entrada (scanf)

- ✓ O comando para leitura de dados do teclado é o **scanf**.
- ✓ Sintaxe:
  - **scanf**(" %[formato]" , &[variável]);
- ✓ Onde:
  - [formato] – define o tipo da variável a ser impressa
  - [variável] – deve ser uma variável de tipo compatível
- ✓ Não se esqueça do **&** antes de ler a variável

# Laboratório de Programação

## Variáveis e Comandos em C

### Comando de Entrada (scanf)

#### ✓ Formatos.

- **%d** : inteiro decimal com sinal
- **%u** : inteiro decimal sem sinal
- **%i** : inteiro decimal, hexadecimal ou octal
- **%f** : ponto flutuante em decimal
- **%ld** : inteiro decimal longo
- **%lf** : ponto flutuante longo
- **%c** : caractere simples
- **%s** : cadeia de caracteres (strings)

# Laboratório de Programação

## Variáveis e Comandos em C

### Comando de Entrada (scanf) - Exemplo

```
#include<stdio.h>
#include<stdlib.h>
int main ( )
{
    float n1, n2, n3;
    scanf("%f" , &n1);
    scanf("%f" , &n2);
    scanf("%f" , &n3);
    scanf(" %f %f %f " , &n1, &n2, &n3);
    system("pause");
    return (0);
}
```

Não esquecer quando for  
ler uma variável do  
símbolo &

# Laboratório de Programação

## Variáveis e Comandos em C

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int main ( )
```

```
{
```

```
    float n1, n2, n3;
```

```
    printf("Informe a primeira nota");
```

```
    scanf("%f" , &n1);
```

```
    printf("Informe a segunda nota");
```

```
    scanf("%f" , &n2);
```

```
    printf("Informe a terceira nota");
```

```
    scanf("%f" , &n3);
```

```
    printf("Os valores são: %f - %f - %f", n1, n2, n3);
```

```
    printf("Informe os três valores");
```

```
    scanf(" %f %f %f " , &n1, &n2, &n3);
```

```
    printf("Os novos valores são: %f - %f - %f", n1, n2, n3);
```

```
    system("pause");
```

```
    return (0); }
```



# Laboratório de Programação

## Variáveis e Comandos em C

### Comando de Saída (printf)

- ✓ O comando para impressão de dados em tela é o **printf**.
  - ✓ Sintaxe:
    - **printf**(" %[formato]" , [variável]);
  - ✓ Onde:
    - [formato] – define o tipo da variável a ser impressa
    - [variável] – deve ser uma variável, constante ou valor de um tipo compatível com o formato
- ↖ Não tem o símbolo &

# Laboratório de Programação

## Variáveis e Comandos em C

### Comando de Saída (printf)

#### ✓ Formatos.

- **%d** : inteiro
- **%f** : ponto flutuante
- **%.2f** : ponto flutuante com 2 casas decimais
- **%lf** : ponto flutuante longo
- **%.2lf** : ponto flutuante longo com 2 casas decimais
- **%c** : caractere simples
- **%s** : cadeia de caracteres (strings)

# Laboratório de Programação

## Variáveis e Comandos em C

### Comando de Saída (printf) - Exemplo

```
#include<stdio.h>
#include<stdlib.h>
int main ( )
{
    int n1, n2;

    scanf("%d",&n1) ;
    printf("O primeiro valor: %d\n", n1) ;
    scanf("%d",&n2) ;
    printf("O primeiro valor %d Segundo %d\n",n1,n2) ;
    printf("Os valores s%co: %d %d\n",134, n1, n2) ;

    system("pause") ;
    return (0) ;
}
```

# Laboratório de Programação

## Variáveis e Comandos em C

### Comando de Saída (printf)

- ✓ Além de escrevermos o conteúdo de variáveis, podemos usar a instrução **printf** para escrever o resultado de expressões diretamente, **sem a necessidade de atribuição**.
- ✓ **Sintaxe:**

```
printf("A media é: %.2f", (a+b)/2);
```

```
printf("O produto dos Números é: %d", a*b);
```

```
printf("Nome: %s , nota: %.2f", nome, nota);
```

# Laboratório de Programação

## Variáveis e Comandos em C

### Transformando um Pseudocódigo em linguagem C

Algoritmo\_Soma

Declare N1, N2, M : Inteiro;  
Escreva ("Digite o primeiro número");  
Leia (N1);  
Escreva ("Digite o segundo número");  
Leia (N2);  
 $M \leftarrow N1 + N2$ ;  
Escreva ("O resultado é: ", M)



```
#include<stdio.h>
#include<stdlib.h>
int main () {
    int N1, N2, M;
    printf ("Digite o primeiro numero\n");
    scanf ("%d",&N1);
    printf("\nDigite o segundo numero");
    scanf("%d",&N2);
    M = (N1 + N2);
    printf ("\nO resultado %c:  %d \n\n", 130,M);
    system("pause");
    return (0);
}
```

Fim\_Algoritmo



# Laboratório de Programação

## Variáveis e Comandos em C

### Explorando a função `printf()`

```
#include<stdio.h> // Para printf()
#include<stdlib.h> // Para system()
int main () {
    int reajuste = 10;
    printf("O reajuste foi de %d%%. \n", reajuste); // Impressão do %
    system("pause");
    return (0);
}
```

# Laboratório de Programação

## Variáveis e Comandos em C

### Explorando a função `printf()`

```
/* Tamanho de campo com inteiros */  
  
#include<stdio.h>  
#include<stdlib.h>  
  
int main () {  
    printf("Os alunos sao %2d. \n", 350);  
    printf("Os alunos sao %4d. \n", 350);  
    printf("Os alunos sao %7d. \n", 350);  
    system("pause");  
    return (0);  
}
```

# Laboratório de Programação

## Variáveis e Comandos em C

### Explorando a função `printf()`

```
/* Tamanho de campo com inteiros */
```

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int main () {
```

```
    int lapis=45, borrachas=2345, canetas=420, cadernos=8, fitas=13050;
```

```
    printf("\nLapis          %12d", lapis);
```

```
    printf("\nBorrachas       %12d", borrachas);
```

```
    printf("\nCanetas          %12d", canetas);
```

```
    printf("\nCadernos          %12d", cadernos);
```

```
    printf("\nFitas              %12d", fitas);
```

```
    system("pause");
```

```
    return (0);
```

```
}
```

# Laboratório de Programação

## Variáveis e Comandos em C

### Explorando a função `printf()`

*/\* Tamanho de campo com ponto flutuante \*/*

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int main () {
```

```
    float lapis=4.875, borrachas=234.542, canetas=42.036, cadernos=8.0,  
    fitas=13.05;
```

```
    printf("\nLapis           %12.2f", lapis);
```

```
    printf("\nBorrachas        %12.2f", borrachas);
```

```
    printf("\nCanetas           %12.2f", canetas);
```

```
    printf("\nCadernos          %12.2f", cadernos);
```

```
    printf("\nFitas             %12.2f \n\n", fitas);
```

```
    system("pause");
```

```
    return (0);
```

```
}
```

# Laboratório de Programação

## Variáveis e Comandos em C

### Exemplo 1 - Programa em C

```
/* Calcula o dobro e o quadrado de um número */  
#include<stdio.h>  
#include<stdlib.h>  
int main () {  
    int A, Dobro, Quadrado;  
    printf("\nDigite um número: \n");  
    scanf("%d", &A);  
    Dobro = A * 2;  
    Quadrado = A * A;  
    printf("O dobro do numero é: %d \n\n", Dobro);  
    printf("O quadrado do numero é: %d \n\n", Quadrado);  
    system("pause");  
    return (0);  
}
```



# Laboratório de Programação

## Variáveis e Comandos em C

### Exemplo 2 - Programa em C

```
/* Calcula a sua idade em dias */  
#include<stdio.h>  
#include<stdlib.h>  
int main () {  
    float anos, dias;  
    printf("Digite a sua idade em anos: ");  
    scanf("%f", &anos);  
    dias = anos * 365;  
    printf("A sua idade em dias é %.0f. \n", dias);  
    system("pause");  
    return (0);  
}
```

# Laboratório de Programação

## Variáveis e Comandos em C

### Exemplo 3 - Programa em C

/\* Mostra o uso do scanf() com várias entradas

Calcula a media de 4 notas \*/

```
#include<stdio.h>
```

```
#include<stdlib.h>
```

```
int main () {
```

```
    float p1, p2, p3, p4;
```

```
    double media;
```

```
    printf("\nDigite as notas das 4 provas: \n");
```

```
    scanf("%f %f %f %f", &p1, &p2, &p3, &p4);
```

```
    media = (p1 + p2 + p3 + p4) / 4.0;
```

```
    printf("\n MEDIA: %.2f \n", media);
```

```
    system("pause");
```

```
    return (0);
```

```
}
```

### Exercícios – Construir os seguintes programas:

- 1 – Faça um algoritmo que leia três números, mostre o resultado da multiplicação dos dois primeiros e somar o resultado da multiplicação com o terceiro número:
- 2 – Faça um programa para mostrar o resultado da divisão de dois números:
- 3 – Faça um programa para somar dois números inteiros e multiplicar o resultado pelo primeiro número e mostre o resultado:
- 4 – Faça um programa que leia um número inteiro e escreva o seu sucessor:
- 5 – Faça um programa que leia 4 variáveis A,B,C e D. A seguir, calcule e mostre a diferença do produto de A e B pelo produto de C e D:
- 6 – Faça um programa que leia duas variáveis A e B. A seguir, calcule o produto entre elas e atribua à variável PROD. A seguir mostre o resultado da variável PROD:





# LABORATÓRIO DE PROGRAMAÇÃO

## Variáveis e Comandos em C

Profº. Sérgio Roberto Costa Vieira, M.Sc.

Cursos de Computação

1º. Período