

### Linguagem C

Projetada por Dennis M. Ritchie, em 1972, no laboratório da Bell.

Em 1973, Dennis M. Ritchie e Ken Thompson, escreveram uma versão do sistema operacional Linux utilizando a linguagem C.

### main()

Principal função. Marca o ponto de inicialização do processo de execução do programa.

### stdio.h

Biblioteca padrão. Inclui as funções printf, scanf, getchar, puts, gets, entre outras.

### função printf()

Utilizada para saída de dados (no vídeo).

Códigos de formatação

%d Permite a escrita de números inteiros (base 10)

%f Permite a escrita de números reais (ponto flutuante)

%c Permite a escrita de apenas um caractere

%s Permite a escrita de uma série de caracteres (string)

```
#include "stdio.h"
main()
{
    int a = 10;
    int b = 20;
    int x = a + b;
    printf("%d", x);
}
```

### função scanf()

Utilizada para entrada de dados (via teclado).

```
#include "stdio.h"
main()
{
    int a;
    int b;
    int x;
    scanf("%d", &a);
    scanf("%d", &b);
    x = a + b;
    printf("%d", x);
}
```

```
#include "stdio.h"
main()
{
    int a;
    int b;
    int x;
    printf("Informe o primeiro valor ... ");
    scanf("%d", &a);
    printf("Informe o segundo valor .... ");
    scanf("%d", &b);
    x = a + b;
    printf("%d", x);
}
```

```
#include "stdio.h"
main()
{
    int a;
    int b;
    int x;
    printf("Soma dois numeros inteiros \n");
    printf("Informe o primeiro valor ... ");
    scanf("%d", &a);
    printf("Informe o segundo valor .... ");
    scanf("%d", &b);
    x = a + b;
    printf("%d", x);
}
```

### Exercício

```
#include "stdio.h"
/*
ht = horas trabalhadas
vh = valor por hora
pd = percentual de desconto
---
sb = salario bruto
vd = valor do desconto
sl = salario liquido

*/
main()
{
    float ht, vh, pd, sb, vd, sl;

    printf("Horas trabalhadas: ..... ");
    scanf("%f", &ht);
    printf("Valor por hora: ..... ");
    scanf("%f", &vh);
    printf("Percentual de desconto: .... ");
    scanf("%f", &pd);

    sb = ht * vh;
    vd = (pd/100) * sb;
    sl = sb - vd;

    printf("Salario Bruto: ..... %7.2f\n", sb);
    printf("Valor Desconto: ..... %7.2f\n", vd);
    printf("Salario Liquido: ..... %7.2f\n", sl);

}
```

### Como resolver problemas com acentuação em português

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

int main(void)
{
    setlocale(LC_ALL, "Portuguese");
    printf("Alô mundo! \n\n");
}
```

### Estruturas de decisão

#### Operadores relacionais

== != < > <= >=

#### Operadores lógicos

&& || !

```
#include "stdio.h"
#include <locale.h>
main()
{
    setlocale(LC_ALL, "Portuguese");
    int a;
    int b;
    printf("Verifica qual número inteiro é maior \n");
    printf("Informe o valor para a: ");
    scanf("%d", &a);
    printf("Informe o valor para b: ");
    scanf("%d", &b);
    if (a > b)
        printf("a é maior que b");
}
```

```
#include "stdio.h"
#include <locale.h>
main()
{
    setlocale(LC_ALL, "Portuguese");
    int a;
    int b;
    printf("Verifica qual número inteiro é maior \n");
    printf("Informe o valor para a: ");
    scanf("%d", &a);
    printf("Informe o valor para b: ");
    scanf("%d", &b);
    if (a > b)
        printf("a é maior que b");
    else if (b > a)
        printf("b é maior que a");
}
```

```
#include "stdio.h"
#include <locale.h>
main()
{
    setlocale(LC_ALL, "Portuguese");
    int a;
    int b;
    printf("Verifica qual número inteiro é maior \n");
    printf("Informe o valor para a: ");
    scanf("%d", &a);
    printf("Informe o valor para b: ");
    scanf("%d", &b);
    if (a > b)
        printf("a é maior que b");
    else if (b > a)
        printf("b é maior que a");
    else
        printf("a e b são iguais");
}
```

```
#include<stdio.h>
//Imprime x vale 10 e y vale 20
void main()
{
    int x=10, y=20;

    if(x == 10)
    {
        printf("x vale 10 ");
        if(y == 20)
        {
            printf("y vale 20");
        }
    }
}
```

```
#include<stdio.h>
//Imprime x vale 10
void main()
{
    int x=10, y=30;

    if(x == 10)
    {
        printf("x vale 10 ");
        if(y == 20)
        {
            printf("y vale 20");
        }
    }
}
```

```
#include<stdio.h>
//Não imprime nada, pois a primeira condição é falsa
void main()
{
    int x=30, y=20;

    if(x == 10)
    {
        printf("x vale 10 ");
        if(y == 20)
        {
            printf("y vale 20");
        }
    }
}
```

```
#include<stdio.h>
//Imprime x vale 10 e vale 20
void main()
{
    int x=10, y=20;

    if(x==10 && y==20)
    {
        printf("Verdadeiro");
    }
}
```

**\*\* Demonstrar com ||**

```
#include "stdio.h"
#include <locale.h>
main()
{
    setlocale(LC_ALL, "Portuguese");
    float a;
    float b;
    float media;
    printf("Calcula a média com base em duas notas \n");
    printf("Informe a 1a nota: ");
    scanf("%f", &a);
    printf("Informe a 2a nota: ");
    scanf("%f", &b);
    media = (a + b) / 2;
    printf("Média: %4.2f", media);
    if (media >= 6)
        printf("\nAPROVADO");
    else
        printf("\nREPROVADO");
}
```

### Estruturas de repetição

#### while

Efetua um teste lógico no início de um looping.

Executa um conjunto de instruções enquanto a condição verificada for verdadeira.

```
/* while */
#include "stdio.h"
main()
{
    int i;
    i = 1;
    while (i <= 3)
    {
        printf("\ni = %d", i);
        i++;
    }
}
```

```
/* while */
#include "stdio.h"
main()
{
    int a, b, s, i;
    i = 1;
    while (i <= 3)
    {
        printf("\n\nInforme um valor (inteiro) para a: ");
        scanf("%d", &a);
        printf("Informe um valor (inteiro) para b: ");
        scanf("%d", &b);
        s = a + b;
        printf("\ni = %d", i);
        printf("\nResultado: %d", s);
        i++;
    }
}
```



### do ... while

Efetua um teste lógico no final de um looping.

Executa um conjunto de instruções pelo menos uma vez antes de verificar a validade da condição estabelecida.

```
/* do ... while */
#include "stdio.h"
main()
{
    int i;
    i = 1;
    do
    {
        printf("\ni = %d", i);
        i++;
    }
    while (i <= 3);
}
```

```
/* do ... while */
#include "stdio.h"
main()
{
    int a, b, s, i;
    i = 1;
    do
    {
        printf("\n\nInforme um valor (inteiro) para a: ");
        scanf("%d", &a);
        printf("Informe um valor (inteiro) para b: ");
        scanf("%d", &b);
        s = a + b;
        printf("\ni = %d", i);
        printf("\nResultado: %d", s);
        i++;
    }
    while (i <= 3);
}
```

### for

Executa a inicialização e testa a condição.

```
/* for */
#include "stdio.h"
main()
{
    int i;
    for (i=1;i<=3;i++)
        printf("i = %d\n", i);
}
```

```
/* for */
#include "stdio.h"
main()
{
    int a, b, s, i;
    for (i=1;i<=3;i++)
    {
        printf("\n\nInforme um valor (inteiro) para a: ");
        scanf("%d", &a);
        printf("Informe um valor (inteiro) para b: ");
        scanf("%d", &b);
        s = a + b;
        printf("\ni = %d", i);
        printf("\nResultado: %d", s);
    }
}
```

```
/* for */
#include "stdio.h"
main()
{
    int i, j;
    printf("Informe a quantidade de repeticoes: ");
    scanf("%d", &j);
    for (i=1; i<=j; i++)
        printf("i = %d\n", i);
}
```

### Matrizes

Matrizes são coleções indexadas de variáveis de mesmo tipo.

#### Matrizes de uma dimensão (Vetores)

Matrizes unidimensionais ou vetores apresentam apenas uma linha e várias "colunas".

*// Exemplo com variáveis escalares:*

```
#include "stdio.h"

main()
{
    float nota1, nota2, nota3, nota4;
    float soma = 0, media;
    printf("\nCalculo de media\n");
    printf("\nInforme as 4 notas\n");
    scanf("%f %f %f %f", &nota1, &nota2, &nota3, &nota4);
    soma = nota1 + nota2 + nota3 + nota4;
    media = soma / 4;
    printf("A media e igual a: %4.2f", media);
}
```

*// Exemplo com vetor*

```
#include "stdio.h"

main()
{
    float notas[4]; // vetor com quatro elementos do tipo float
    // 1° elemento notas[0], 2° elemento notas[1], ...
    float soma = 0, media;
    int i;
    printf("\nCalculo de media\n\n");
    for (i = 0; i <= 3; i++) {
        printf("Informe a %da. nota: ", i+1);
        scanf("%f", &notas[i]);
        soma += notas[i];
    }
    media = soma / 4;
    printf("A media e igual a: %4.2f", media);
}
```

### Matrizes de duas ou mais dimensões

Matrizes unidimensionais ou vetores apresentam apenas uma linha e várias "colunas".

```
// Exemplo com matrizes multidimensionais
#include "stdio.h"
main()
{
    float notas[8][4];
    float numero;
    int i, j;
    printf("\nCalculo de media\n\n");
    for (i = 0; i <= 7; i++) {
        printf("Informe a nota do %do. aluno: ", i+1);
        for (j = 0; j <= 3; j++) {
            printf("Nota %d: ", j+1);
            scanf("%f", &numero);
            notas[i][j] = numero;
        }
    }

    // Saida de notas

    for (i = 0; i <=7; i++) {
        printf("\nAs notas do aluno %d sao: \n\n", i+1);
        for (j = 0; j <= 3; j++) {
            printf("Nota %d: %5.2f\n", j+1, notas[i][j]);
        }
    }
}
```

### Strings

String, na linguagem C, é uma matriz unidimensional do tipo **char**.

Visto que String é uma matriz, então é possível acessar qualquer um dos seus elementos através dos seus respectivos índices.

```
#include "stdio.h"

main() {
    char nome[10];
    printf("Digite o seu nome: ");
    scanf("%s", &nome); // ou: scanf("%s", nome);
    printf("Você digitou: %s", nome);
}
```

```
#include "stdio.h"

main() {
    char nome[10];
    puts("Digite o seu nome: ");
    gets(nome);
    printf("Você digitou: %s", nome);
}
```

```
//percorrendo um vetor de char
#include <stdio.h>

main() {
    int i;
    char texto[] = "string";

    printf("Valor da variavel texto = %s\n", texto);

    for (i=0; i<6; i++) {
        printf("Valor do elemento %d da string = %c\n",i, texto[i]);
    }
}
```

### Estruturas (Matrizes Heterogêneas)

Matrizes trabalham com um único tipo de dado.

Estruturas em C (também chamadas de registro, em outras linguagens) podem trabalhar com tipos de dados diferentes. Cada elemento de um registro costuma ser denominado de campos.

```
#include "stdio.h"

main() {
    struct cad_aluno {
        char nome[30];
        float a1;
        float a2;
        float a3;
        float a4;
    };

    struct cad_aluno aluno;

    printf("Cadastro de Aluno\n\n");
    printf("Informe o nome .....: "); scanf("%s", aluno.nome);
    printf("Informe a 1a. nota .....: "); scanf("%f", &aluno.a1);
    printf("Informe a 2a. nota .....: "); scanf("%f", &aluno.a2);
    printf("Informe a 3a. nota .....: "); scanf("%f", &aluno.a3);
    printf("Informe a 4a. nota .....: "); scanf("%f", &aluno.a4);
    printf("\n");
    printf("Nome .....: %s\n", aluno.nome);
    printf("Nota 1 .....: %4.2f\n", aluno.a1);
    printf("Nota 2 .....: %4.2f\n", aluno.a2);
    printf("Nota 3 .....: %4.2f\n", aluno.a3);
    printf("Nota 4 .....: %4.2f\n", aluno.a4);

}
```