

Introdução a Linguagem C (Parte I)

UFPA – Sistemas de Informação

**Roberto Araujo
2013**

Meu primeiro programa em C

```
#include <stdio.h>
main() {
    printf("Primeiro Programa \n");
}
```

← Biblioteca padrão

← Função principal

← Função na biblioteca padrão

Início e fim da função

O C faz distinção entre letras maiúscula e minúsculas

Tipos de Dados Básicos em C

- **int** – Inteiro
 - O sinal de menos indica um valor negativo
- **float** – Real
- **double** – Real
 - Similar ao float, mas possui uma precisão duas vezes maior
- **char** – caracter
 - Utilizado para armazenar um único caracter

Ler e Escrever

- Escrevendo dados

```
printf("xxx");  
printf("xxx %i", var_inteira);  
printf("xxx %f", var_float);
```

- Leitura de Dados;

```
scanf("%i", &var_inteira);  
scanf("%f", &var_float);  
scanf("%f", &var_float);  
scanf("%c", &var_caracter);  
scanf("%i:%i:%i", &hora, &minuto, &segundo);
```

Ex:

```
#include <stdio.h>  
  
main() {  
    int idade;  
    printf("Informe sua idade: \n");  
    scanf("%i", &idade);  
    printf("A idade informada é: %i  
    \n", idade);  
}
```

Exemplo

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

```
main() {
```

```
    int num1 = 100;
```

```
    float num2 = 50.15;
```

```
    double num3 = 50.15;
```

```
    char letra = 's';
```

```
    printf("Inteiro = %i \n", num1);
```

```
    printf("Real float = %f \n", num2);
```

```
    printf("Real double = %g\n", num3);
```

```
    printf("Caractere = %c \n", letra);
```

```
}
```

Expressões Aritméticas

```
#include <stdio.h>
main() {
    int a = 50;
    int b = 5;
    int c = 25;
    int d = 3;
    int result;
    float e = 2.0;

    result = a - b;
    printf("a - b = %i \n", result);
    result = b * c;
    printf("b * c = %i \n", result);
    result = a / b;
    printf("a / b = %i \n", result);
    result = a + b * c;
    printf("a + b * c = %i \n", result);

    printf("a * b + c * d = %i \n", a * b + c * d);
    printf("-a = %i", -a);
    printf("c / d * d = %f\n", c / d * d);
}
```

Soma: +

Subtração: -

Multiplicação: *

Divisão: /

Resto da divisão: %

Operadores Relacionais

Operador	Descrição	Exemplo
==	Igualdade	x == 10
!=	Diferente	y != 5
<	Menor	x < y
<=	Menor igual	baixo <= alto
>	Maior	a > b
>=	Maior igual	c >= 0

Laço (Loop) - Para

for (exp_inicial; cond_do_laço; exp_do_laço)

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

```
main() {
```

```
    int n, num_tri;
```

```
    num_tri = 0;
```

```
    for ( n=1; n<=200; n = n+1) {
```

```
        num_tri = num_tri + n;
```


```
    }
```

```
    printf("O 200 número triangular é %i \n", num_tri);
```

```
}
```


Laço (Loop) - Enquanto

```
while ( cond_laço ) {  
    declarações_do_programa  
    expressão_do_laço;  
}
```

```
#include <stdio.h>  
main() {  
    int cont = 1;  
    while ( cont <= 5) {  
        printf("%i \n", cont);  
        cont = cont + 1;  
    }  
    }  cont++;
```

Laço (Loop) - Repita-Ate

- do-while

do {

declarações_do_programa

} while (expressão);

```
/* Reverte os digitos de um número */
#include <stdio.h>
main() {
    int num, dig_direita;

    printf("Informe um número: ");
    scanf("%i", &num);
    do {
        digito_direita = num % 10;
        printf("%i", dig_direita);
        num = num / 10;
    } while ( num != 0);
    printf("\n");
}
```

O 'num' é um inteiro e um valor real pode ser obtido de num/10.

O C trunca o número (a parte fracional é descartada) e o converte para inteiro

Estrutura de Decisão

- Se-Então (if)

```
if ( expressão) {  
    declarações_do_programa  
}
```

```
/* Calcula o valor absoluto de um inteiro */  
#include <stdio.h>  
main() {  
    int num;  
  
    printf("Informe um número: ");  
    scanf("%i", &num);  
    if ( num < 0) {  
        num = -num;  
    }  
    printf("O valor absoluto de %i \n é", num);  
}
```

Estrutura de Decisão

- Se-Então-Senão (if-else)

```
if ( expressão ) {  
    declarações_do_programa 1  
} else {  
    declarações_do_programa 2  
}
```

```
/* Determina se um número é par ou ímpar */  
#include <stdio.h>  
main() {  
    int testnum, resto;  
  
    printf("Informe o número a ser testado: ");  
    scanf("%i", &testnum);  
  
    resto = testnum % 2;  
  
    if ( resto == 0 )  
        printf("O número é par");  
    else  
        printf("O número é ímpar");  
}
```

Operador
de módulo em C



Operadores Lógicos

Operador	Descrição	Exemplo
&&	e	(x>0) && (y<5)
	ou	(z > 10.0) (z < -5.0)
!	não	!(altura == 0.0)

Estruturas de Decisão (aninhadas)

- Ex 1:

```
if ( jogo == 0)
    if ( mov == 1)
        printf("movimento para direita");
    else
        printf("movimento para esquerda);
```

- Ex 2:

```
if ( fim_jogo == 0)
    if ( mov == 1)
        printf("movimento para direita");
    else
        printf("movimento para esquerda);
else
    printf("Final de jogo");
```

Estruturas de Decisão

- else if

```
if ( expressão 1)
    declaração_do_programa 1
else
    if ( expressão 2 )
        declaração_do_programa 2
    else
        declaração_do_programa 3
```

-

```
if ( expressão 1)
    declaração_do_programa 1
else if ( expressão 2 )
    declaração_do_programa 2
else
    declaração_do_programa 3
```

```
/* Implementa a função sinal */
#include <stdio.h>
main() {
    int num, sinal;
    printf("Informe um número: ");
    scanf("%i", &testnum);
    if ( num < 0)
        sinal = -1;
    else if ( num == 0)
        sinal = 0;
    else
        sinal = 1;
    printf("Sinal = %i \n", sinal);
}
```

Exercícios

- Faça um programa que, dado um número inteiro positivo n , o programa deve imprimir os números naturais pares até n .
- Faça um programa em que o usuário informe um valor inteiro e o programa retorne o fatorial desse número. Caso o valor seja negativo, o programa deve solicitar ao usuário um novo valor.
- Faça um programa em que o usuário informa um número inteiro. O programa deve então extrair e mostrar cada dígito do número em português. (desafio)
- Escreva um programa para calcular o valor de um número inteiro (base 10) de acordo com a base (binário, octal, hexadecimal) informada pelo usuário.

Vetores



O primeiro elemento do vetor é indexado por 0

O último elemento é indexado pelo número de elementos -1

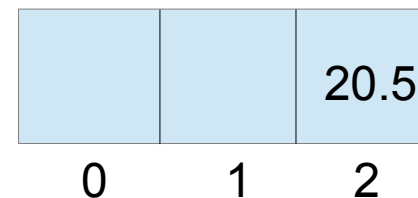
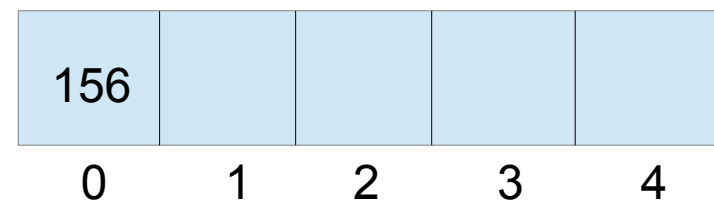
Reserva um conjunto de espaços de armazenamento na memória, de um mesmo tipo.

Declarando vetores: **tipo variavel[num_de_posições];**

Ex: `int casa[5];`
`float temp[3];`

Atribuindo valores:

`casa[0] = 156;`
`temp[2] = 20.5;`



Exemplo

```
#include <stdio.h>

main() {
    int val[10];
    int i;
    val[0] = 197;
    val[2] = -100;
    val[5] = 350;
    val[3] = val[0] + val[5];
    val[9] = val[5] / 10;
    --val[2];
    for ( i = 0; i<10; i++)
        printf("valor[%i] = %i \n", i, val[i]);
}
```

Exercício

- Escreva um programa para calcular a tabuada de 5 armazenando os valores em um vetor
- Escreva um programa em que o usuário entre com 10 valores e seja calculado a média desses valores. O programa deve apresentar ao usuário a média obtida.
- Escreva um programa para calcular os 10 primeiros números de Fibonacci, armazenando-os em um vetor. O vetor deve ser então apresentado ao usuário.

Vetores

- Inicializando um Vetor

```
int cont[5] = { 0, 1, 2, 3, 4};
```

```
char letras[5] = { 'a', 'b', 'c', 'd', 'e' };
```

```
float temp[40] = { 30.0, 25.2, 42.0 }; —————▶ Inicia os primeiros 3 valores.
```

O restante da lista é
inicializado com ZERO

Vetores Multidimensionais

- A linguagem C permite a definição de vetores de qualquer dimensão
- **Matrizes = Vetor de duas dimensões**

$$\begin{pmatrix} 1 & 2 & 3 \\ 6 & 7 & 9 \\ 5 & 8 & 10 \end{pmatrix}_{3 \times 3}$$

Declarando matrizes: **tipo var[num_pos][num_pos];**

Ex: int matriz[3][3];

Matrizes

- Inicializando uma matriz

$$\begin{pmatrix} 1 & 2 & 3 \\ 6 & 7 & 9 \\ 5 & 8 & 10 \end{pmatrix}_{3 \times 3}$$

```
int M[3][3] = {  
    { 1, 2, 3},  
    { 6, 7, 9},  
    { 5, 8, 10}  
};
```

Exercício

- Escreva um programa em que o usuário entre com as notas de 5 alunos em 3 disciplinas. O programa deve então calcular a média de cada aluno e apresentar a maior média.
- Escreva um programa para encontrar os números primos até 150 através do algoritmo “**Crivo de Eratosthenes**” apresentado a seguir:
 1. Definir um vetor de inteiros 'P' (matriz $P_{i \times j}$). Atribuir 0 para todos os elementos 'P' ($P_{i,j}$). Onde: $2 \leq i \leq n$ ($2 \leq i,j \leq n$)
 2. Atribuir i para 2
 3. Se $i > n$, o algoritmo termina
 4. Se ' P_i ' ($P_{i,j}$) for 0, então 'i' é primo
 5. Para todos os valores positivos de j, tal que ' $i \times j < n$ ', atribua ' $P_{i \times j}$ ' para 1.
 6. Adicione 1 em i e retorne ao passo 3

Exercícios

- Faça um programa para criar a transposta de uma matriz. O usuário deve informar a matriz.
- Faça um programa que calcule a soma dos valores da diagonal principal de uma matriz 5x5. O usuário deve informar a matriz.
- Faça um programa para calcular a multiplicação de duas matrizes 3x3. O usuário deve entrar com as duas matrizes e o programa deve armazenar o resultado em uma terceira matriz.

Exercício

- Escreva um programa em que o usuário entra com um número 'n' e o programa retorna os números primos até 'n'. Os números primos encontrados devem ser armazenados em um vetor
- Escreva um programa em que o usuário entre com um número positivo e o programa converta esse número para base 2. O número convertido para a base 2 deve ser armazenado em um vetor.
- Escreva um programa em que o usuário entre com um número em decimal e um base (binária, octal ou hexadecimal). O algoritmo deve então armazenar o número convertido em um vetor e apresentar o vetor.