



Laboratório de Computação I

Função e Passagem de Parâmetro



Prof. Ivre Marjorie

Introdução

- ▶ Nessa aula vamos praticar os tipos de funções e passagem de parâmetros (por valor e por referência).
- ▶ Uma função é um conjunto de instruções desenhadas para cumprir uma tarefa particular e agrupadas numa unidade com um nome para referencia-la.
- ▶ Toda função tem um tipo, nome e pode ter ou não parâmetros.

```
TipoDaFunção NomeDaFunção( Parâmetros )  
{  
    instruções;  
}
```

Introdução

- ▶ É possível construir as seguintes funções:
 1. Função sem retorno e sem parâmetro
 2. Função sem retorno e com parâmetro
 3. Função com retorno e sem parâmetro
 4. Função com retorno e com parâmetro
- ▶ Funções sem retorno serão do tipo **void**
- ▶ Funções com retorno podem ser de qualquer tipo já visto (**double, int, float, char, etc**)
- ▶ Os parâmetros podem ser passados por valor (copia) e por referência (acesso ao mesmo endereço)





Exemplo 1

- ▶ Função **sem** retorno e **com** parâmetro

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
void verificaNum(int);
int main()
{
    int num;
    printf("Digite um numero: ");
    scanf("%d", &num);
    verificaNum(num);
    return 0;
}
```

```
void verificaNum(int numero)
{
    char tipo[10];
    if (numero < 0)
        strcpy(tipo, "negativo");
    else if (numero > 0)
        strcpy(tipo, "positivo");
    else
        strcpy(tipo, "nulo");
    printf("O numero %d e: %s", numero, tipo);
    return tipo;
}
```





Exemplo 2

► Função com retorno e com parâmetro

```
double calculaMedia(double, double);
int main()
{
    double num1, num2;
    printf("Digite um numero: ");
    scanf("%lf", &num1);
    printf("Digite outro numero: ");
    scanf("%lf", &num2);
    printf("A media de %.2lf e %.2lf e: %.2lf", num1, num2, calculaMedia(num1, num2));
    return 0;
}

double calculaMedia(double x, double y)
{
    double media;
    media = (x+y)/2;
    return media;
}
```



Exemplo 3

- Função sem retorno e com parâmetro (por referência)

```
void calculaMedia(double, double, double *);  
int main()  
{  
    double num1, num2, med;  
    printf("Digite um numero: ");  
    scanf("%lf", &num1);  
    printf("Digite outro numero: ");  
    scanf("%lf", &num2);  
    calculaMedia(num1, num2, &med);  
    printf("A media de %.2lf e %.2lf e: %.2lf", num1, num2, med);  
    return 0;  
}  
  
void calculaMedia(double x, double y, double *media)  
{  
    *media = (x+y)/2;  
}
```



Exemplo 4

► Passando vetor como parâmetro

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

#define tam 10
void PreencheVetor(int[]);
void MostraVetor(int[]);
int main()
{
    int vetor[tam];
    PreencheVetor(vetor);
    MostraVetor(vetor);
    return 0;
}
```

```
void PreencheVetor(int vet[])
{
    int i;
    for(i=0;i<tam;i++)
    {
        printf("Digite um numero: ");
        scanf("%d", &vet[i]);
    }
}

void MostraVetor(int vet[])
{
    int i;
    printf("\n\nVetor:");
    for(i=0;i<tam;i++)
    {
        printf(" %d | ", vet[i]);
    }
}
```



Exercícios

1. Faça uma **função** que receba como parâmetro a nota de um aluno e retorne um dos seguintes conceitos:
- ▶ Conceito A - $\text{Nota} \geq 90$;
 - ▶ Conceito B - $90 > \text{Nota} \geq 80$;
 - ▶ Conceito C - $80 > \text{Nota} \geq 70$;
 - ▶ Conceito D - $69 \geq \text{Nota} \geq 60$;
 - ▶ Conceito E - $\text{Nota} < 60$.
- ▶ A nota e o nome do aluno devem ser introduzidos pelo usuário através teclado na função principal. E ao final do processamento o programa deve imprimir na tela a seguinte mensagem:
- O aluno Fulano de Tal obteve nota XX e tem conceito Y**
-





Exercícios

2. Faça um programa que solicite ao usuário uma quantidade inteira de horas. E como resultado deverá ser apresentado, use uma função para cada uma das conversões:
 - ▶ a quantidade de minutos da hora.
 - ▶ a quantidade de segundos da hora.
 - ▶ a quantidade de milissegundos da hora.
3. Faça uma função que receba, por parâmetro, a altura (alt) e o sexo de uma pessoa e retorne o seu peso ideal. Para homens calcular o peso usando a fórmula: **Peso ideal = $72,7 * \text{alt} - 58$** e para mulheres: **Peso ideal = $62,1 * \text{alt} - 44,7$** .





Exercícios

4. Faça uma **função** sem retorno que receba uma matriz **4 x 5** como parâmetro, calcula e mostra um vetor com quatro posições, onde cada posição contém a soma dos elementos de cada linha da matriz. Mostre apenas os elementos do vetor maiores que dez (10). Se não existir nenhum elemento maior que dez mostre uma mensagem. Faça um programa principal que preencha uma matriz **4 x 5** com números inteiros e positivos, em seguida, chame a função.
5. Faça uma **função** que receba um vetor A com 10 números reais, ordene o vetor de forma crescente. Faça outra função para mostrar o vetor ordenado na tela. O vetor deverá ser criado e preenchido na função principal (main).

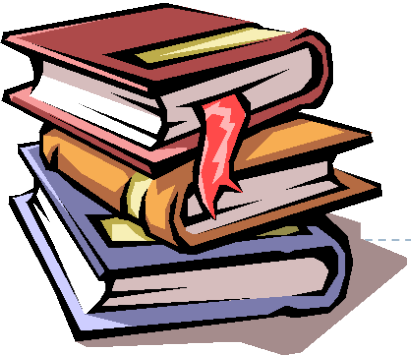




Exercícios

6. Diz-se que um inteiro é primo se for divisível apenas por 1 e por si mesmo. Por exemplo, 2, 3, 5, 7 são primos, mas 4, 6, 8, 9 não são. Escreva uma **função** que determine se um número é primo. Faça um programa que utilize esta função para determinar e imprimir todos os números primos entre 1 e 1000.
7. Faça um programa que preencha uma matriz (6x4), em seguida mostre essa matriz usando uma função **ImprimeMatriz()**, que recebe a matriz como parâmetro. Crie também uma **função** que recebe a matriz e recalcula a matriz digitada, onde cada linha deverá ser multiplicada pelo maior elemento da linha em questão e novamente chame a função para mostrar a matriz na tela.





Referência Bibliográfica

- ▶ MIZRAHI, Victorine Viviane. **Treinamento em linguagem C**. São Paulo: Pearson Prentice Hall, 2008. 2ª edição. Curso Completo.
- ▶ ASCENCIO, Ana Fernanda Gomes e CAMPOS, Edilene A. Veneruchi. **Fundamentos da Programação de Computadores – Algoritmos, Pascal, C/C++ e Java**. São Paulo: Pearson Prentice Hall, 2012. 3ª Edição.

