



Algoritmos e Estruturas de Dados

Funções

Prof^a. Me. Carla Plantier Message
2023

Introdução

- Funções são blocos de código que podem ser nomeados e chamados de dentro de um programa.
 - **printf()**: função que escreve na tela
 - **scanf()**: função que lê o teclado

- É possível ao programador, além disso, escrever suas próprias rotinas. São as chamadas de funções de usuário ou rotinas de usuário.
- Deste modo pode-se segmentar um programa grande em vários programas menores.

- Esta segmentação é chamada de modularização e permite que cada segmento seja escrito, testado e revisado individualmente sem alterar o funcionamento do programa como um todo.

Por que usar funções?

- Para permitir o reaproveitamento de código já construído (por você ou por outros programadores);
- Para evitar que um trecho de código que seja repetido várias vezes dentro de um mesmo programa;
- Com o uso de uma função pode-se alterar apenas dentro da função que se deseja;
- Para que os blocos do programa não fiquem grandes demais;
- Para facilitar a leitura do programa-fonte;
- Para separar o programa em partes (blocos) que possam ser logicamente compreendidos de forma isolada; ou desenvolvido por vários programadores em conjunto.

- **Sintaxe de uma função em C:**

```
tipo_da_funcao NomeDaFuncao (Lista_de_Parametros)
{
    // corpo da função
}
```

- A Lista_de_Parametros, também é chamada de Lista_de_Argumentos, é opcional.

Valor que a
função pode ou
não retornar

Nome que
identifica a
função

Lista de valores
passados para a
função processar

```
tipo_retornado nome_função (parâmetros) {  
    conjunto de declarações e comandos  
}
```

Corpo da função
(tarefas que ela
executa)

- O corpo da função é a sua alma.
 - É formado pelos comandos que a função deve executar
 - Ele processa os parâmetros (se houver), realiza outras tarefas e gera saídas (se necessário)
 - Similar a cláusula **main()**

```
int main() {  
  
    //conjunto de declarações e comandos  
  
    return 0;  
}
```


- A declaração de parâmetros é uma lista de variáveis juntamente com seus tipos:
 - *tipo1 nome1, tipo2 nome2, ... , tipoN nomeN*
 - Pode-se definir quantos parâmetros achar necessários



```
//Declaração CORRETA de parâmetros
int soma(int x, int y){
    return x + y;
}

//Declaração ERRADA de parâmetros
int soma(int x, y){
    return x + y;
}
```

Exemplo

- Faça um programa para apresentar uma mensagem, por meio de uma função:

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

```
void mensagem ( ) { // função sem retorno e sem parâmetro
```

```
    printf (“Boa noite, turma!”);
```

```
}
```

```
main( ){
```

```
    mensagem( ); // chamada da função mensagem
```

```
}
```

Exemplos

- Função para calcular a média de 2 números. (sem retorno e com retorno).

Exercícios

- 1 - Faça um programa por meio de funções que receba 2 números inteiros, calcule e exiba a multiplicação dos números lidos.
- 2 - Faça um programa por meio de funções que leia 2 números inteiros e exiba qual é o maior.
- 3 - Faça um programa por meio de funções que leia uma letra e imprima se a letra é uma vogal.
- 4 - Faça um programa por meio de funções que leia 2 números e exiba a raiz quadrada da soma dos quadrados.

Unoeste

