

Funções

Introdução à Programação

29 de Março de 2018

1 Funções

- Estrutura
- Declaração
- Definição

- A linguagem C nos permite dividir nosso código em funções.
- Dessa forma, podemos criar códigos que são mais reutilizáveis, concisos, bem organizados e eficientes.
- O conceito de funções é extremamente necessário tanto no âmbito matemático quanto no computacional.

- A estrutura de uma função em **C** é a seguinte:

```
tipo do retorno  nome  (parâmetro 1, parâmetro 2, ..., parâmetro n)  
{  
    código  
  
    return...  
}
```

- Da mesma forma que na definição matemática, uma função possui um conjunto de parâmetros de entrada e uma saída.
- Porém, essa função pode ou não retornar um valor.
- Cada *parâmetro* consiste em um tipo e um nome de variável.
- O *tipo de retorno* da função nos diz o que essa função irá retornar.
- Se esse tipo é **void**, a função não retorna valor algum.

- Em **C**, podemos criar tanto a **declaração** de uma função quanto sua **definição**.
- A declaração de uma função indica como a função será chamada e o que ela retornará.
- Consiste somente no *tipo do retorno*, *nome*, e no **tipo** (o nome dos parâmetros é opcional) dos parâmetros.
- Pode aparecer diversas vezes no mesmo programa, desde que todas as definições sejam iguais.

- Exemplo de declarações de funções:

```
int funcao1 (int x);
```

```
int funcao1 (int);
```

```
void funcao2 (double);
```

```
double funcao3 (int , double , char);
```

- A definição de uma função consiste no corpo da função, o que realmente será executado, assim como a sua declaração.
- Pode aparecer somente uma vez no mesmo programa.
- A declaração é incluída nos headers (arquivos .h) e a definição nos arquivos fonte (arquivos .c).

- Exemplo de definições de funções:

```
int funcao1 (int x)
{
    return x * 2;
}
```

```
void funcao2 (double x)
{
    printf("%lf\n", x);
}
```

- Printar o valor da sequência de Fibonacci na posição 1.000.
- Printar o valor da sequência de Fibonacci na posição 10.000.000.
- Dado um vetor de inteiros **ordenado** de tamanho 100.000.000, encontrar um determinado valor em menos de um segundo.