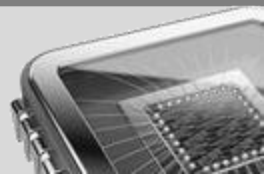


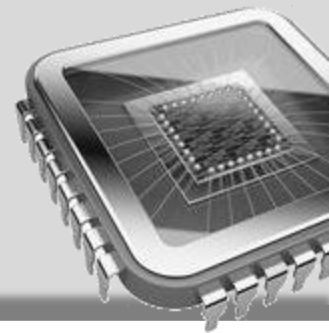
Introdução à Programação

Aula 09

Prof. Max Santana Rolemberg Farias
max.santana@univasf.edu.br
Colegiado de Engenharia de Computação



Funções



- A linguagem C, como qualquer outra linguagem de programação, permite dividir um programa em pedaços (módulos) pequenos e simples.

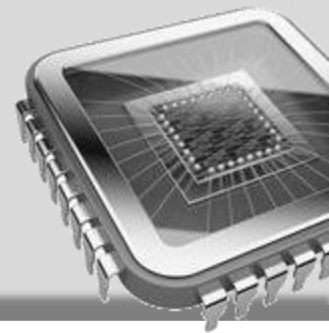
VOCE
SABIA



Embora os programas não modularizados possam resolver a maioria dos problemas, fazer a modularização facilita a vida do programador.

Funções

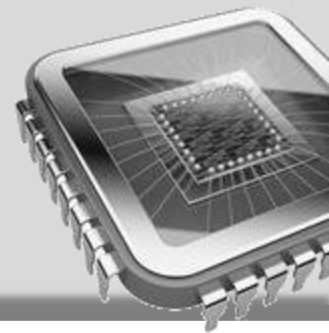
Por que usar funções?



- Para permitir o reaproveitamento de código já construído;
- Para evitar que um trecho de código seja repetido várias vezes dentro de um mesmo programa;
- Para permitir a alteração de um trecho de código de uma forma mais rápida;
- Para que os blocos do programa não fiquem grandes demais e, por consequência, mais difíceis de entender;
- Para facilitar a leitura do programa de uma forma mais fácil;
- Para separar o programa em partes (módulos) que possam ser logicamente compreendidas de forma isoladas

Funções

Estrutura de uma Função

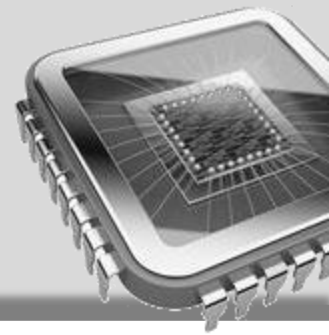


- A estrutura de uma função em C é semelhante à da função *main*. A diferença é que a função *main* possui um **nome especial**.
- O código que descreve o que a função faz é chamado de **definição da função**. Sua forma geral é a seguinte:

```
< tipo > <nome_função>(<parâmetros>) {  
    return <valor>;  
}
```

Funções

Tipos de uma Função



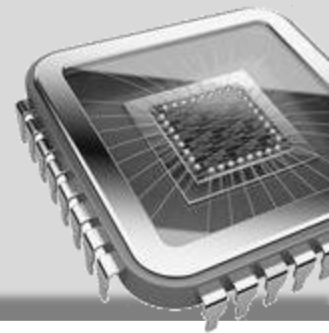
- Os tipos de uma função é definido pelo valor que ela retorna por meio do comando **return**. Uma função é do tipo **int** quando retorna um valor do tipo **int**.



As funções em C podem ter os mesmo tipos das variáveis, exceto quando a função não retorna nada. Nesse caso, ela é do tipo **void**.

Funções

Protótipo de Função



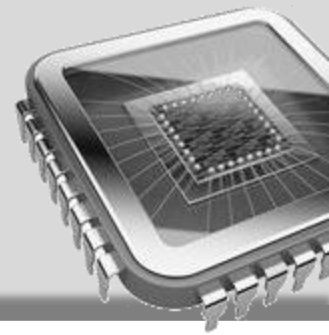
- Uma função não pode ser chamada sem antes ter sido declarada. A declaração de uma função é chamada protótipo da função.
 - O protótipo de função, em geral é colocado no início do programa, que estabelece o tipo da função e os parâmetros (argumentos) que ela recebe



O protótipo de uma função deve preceder sua definição e sua chamada. O protótipo tem a mesma forma da primeira linha da função, exceto por terminara com ponto-e-vírgula.

Funções

Protótipo de Função



- Suponha um programa para calcular a média de um determinado aluno. Para isso, vamos escrever uma função que calcula a média.

```
// protótipo da função
```

```
float media(float nota1, float nota2);
```

```
// Definição da função
```

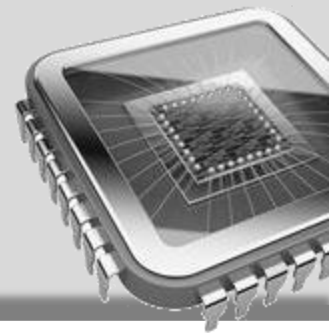
```
float media(float nota1, float nota2) {
```

```
    return (nota1+nota2)/2;
```

```
}
```

Funções

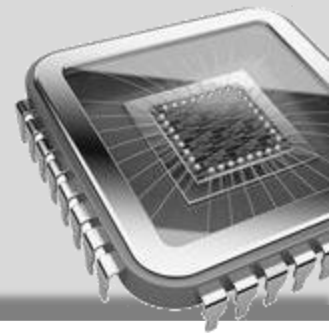
Observações



- O protótipo de uma função deve preceder sua definição e sua chamada.
- As funções definidas antes de serem chamadas não necessitam de protótipo.
- O tipo de uma função é determinado pelo valor que ela retorna via comando `return`, e não pelo tipo dos argumentos que ela recebe.
- Uma função pode não retornar nenhum tipo, essa função é do tipo `void` e não precisa do comando `return`.
- O comando `return` termina a execução da função e retorna o controle para a instrução seguinte do código de chamada.
- O comando `return` pode retornar somente um único valor para a função que chama.

Funções

Parâmetros da Função



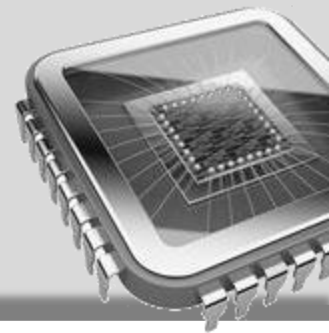
- Parâmetros são valores passados para as funções
 - Existem dois tipos de passagem de parâmetros: passagem por valor e passagem por referência
- Os parâmetros são declarados entre os parênteses da função:

float media(float nota1, float nota2)

Parâmetros

Funções

Passagem por Valor

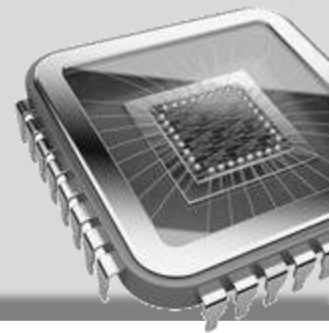


- Na passagem de parâmetro por valor, uma cópia da variável é feita durante a ativação da função.
- O parâmetro é uma variável local dentro da função, totalmente independente dos argumentos da ativação. A passagem de parâmetro por valor funciona como um meio de entrada para a função.

`float media(float nota1, float nota2)`

Funções

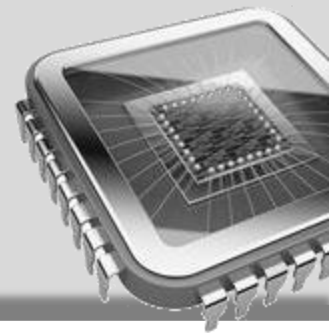
Passagem por Referência



- Na passagem por referência, o endereço da variável, e não uma cópia do valor, é passado durante a ativação da função.
- O que ocorre neste processo é que apesar de ter outro identificador, os valores manipulados na função estão no mesmo espaço de memória dos argumentos e são efetivamente, a mesma variável.

Funções

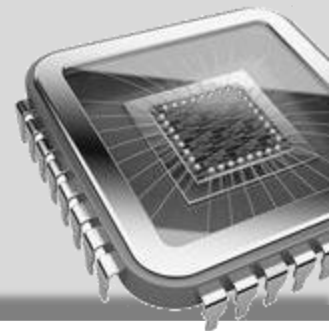
Passagem por Referência



- O uso de parâmetro passado por referência, diferentemente dos parâmetros passados por valor, funciona tanto como entrada para a função como também saída.
 - Quando é necessário que mais de um valor seja retornado por uma função, a utilização de passagem por referência é indicada.
- ```
void media(float nota1, float nota2, float *media)
```

# Funções

## Operador de Referência &



- O operador & é utilizado para obter o endereço de uma variável.
- Referência é um link para uma variável já existente.

```
int x = 5;
```

```
int *y = &x;
```

- A variável y é um link para a variável x. Toda operação em qualquer uma das variáveis tem o mesmo resultado. A referência não é uma cópia da variável a que se refere.