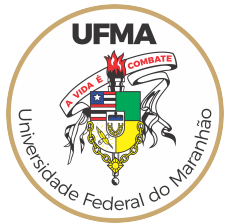
Laboratório de Programação

Prof. Dr. Paulo Rogério de Almeida Ribeiro

Coordenação do Curso de Engenharia da Computação

Funções



Programa

- Coleção de funções;

Programa

- Coleção de funções;
- Função:

Programa

- Coleção de funções;
- Função:
  - Bloco de código;

Programa

- Coleção de funções;
- Função:
  - Bloco de código;
  - Possui nome;

Programa

- Coleção de funções;
- Função:
  - Bloco de código;
  - Possui nome;
  - Pode ter parâmetros de entrada;

Programa

- Coleção de funções;
- Função:
  - Bloco de código;
  - Possui nome;
  - Pode ter parâmetros de entrada;
  - Pode ter retorno (saída).

Função

```
tipo nomeDaFuncao (lista de parâmetros) //opcional
{
        // corpo da função
  return variável_do_mesmo_tipo;
}
```

Função

```
tipo nomeDaFuncao (lista de parâmetros) //opcional
{
        // corpo da função
  return variável_do_mesmo_tipo;
}
```

Nem parâmetros nem return são obrigatórios!

Função

```
tipo nomeDaFuncao (lista de parâmetros) //opcional
{
        // corpo da função
  return variável_do_mesmo_tipo;
}
```

Tipo: tipo de valor que o comando *return* devolve;

Função

```
tipo nomeDaFuncao (lista de parâmetros) //opcional
{
        // corpo da função
  return variável_do_mesmo_tipo;
}
```

Tipo: tipo de valor que o comando *return* devolve;

Parâmetros: tipo e nome das variáveis.

Exemplo de uma função: soma

```
float soma(float a, float b) {
return a+b;
}
```

Exemplo de uma função: soma

```
float soma(float a, float b){
return a+b;
}

void soma(float a, float b){
printf("%f",a+b);
}
```

Exemplo de uma função: soma

```
float soma(float a, float b){
return a+b;
}
```

```
void soma(float a, float b){
printf("%f",a+b);
}
```

Exemplo com e sem retorno

Função

- Função deve ser declarada antes de ser usada;

Função

- Função deve ser declarada antes de ser usada;
- Declaração: tipo NomeDaFuncao (parametros);

Função

- Função deve ser declarada antes de ser usada;
- Declaração: tipo NomeDaFuncao (parametros);
- Chamada a função: nome (função) e parâmetros.

Exemplo completo da função soma

```
#include <stdio.h>

float soma( float a, float b); //protótipo da função, logo após o include

int main(){
float num1, num2, result;
printf ("Num 1:");
scanf("%f", &num1);
printf ("Num 2:");
 scanf("%f", &num2);
result = soma(num1,num2);     //variável recebe o resultado da função
 printf("O resultado é %0.2f", result); return 0;
}

float soma(float a, float b){       //função
return a+b;
}
```

Escopo de variáveis

Escopo de variáveis

- Variável local: apenas dentro da função;

Escopo de variáveis

- Variável local: apenas dentro da função;
- Variável global: código inteiro.

Escopo de variáveis

```c
#include <stdio.h>
int x=20;

void teste1(){
int x=5;
printf("Valor de x dentro da função 1: %d\n", x);
}

void teste2(){
int x=10;
printf("Valor de x dentro da função 2: %d\n", x);
}

int main(){
printf("Valor de x na função main: %d\n", x);
teste1();
teste2();
return 0;
}
```

FUNÇÕES E VETORES

```
#include <stdio.h>
#define TAM 5

void printVec (int x, int vetor[TAM])  // declaracao do vetor
for (int i=0; i< x; i++){
vetor[i] = i;
printf("%d\n", vetor[i]);}
}

int main(){
int vec[TAM];
printVec(TAM, vec); // Passa o vetor como parâmetro
return 0;
}
```

Exercícios

1) Escreva um programa que leia dois números do teclado, bem como uma função que subtraia eles

Exercícios

1) Escreva um programa que leia dois números do teclado, bem como uma função que subtraia eles

2) Implemente uma função que recebe um vetor e um escalar e multiplica cada valor do vetor pelo escalar. A função deve mostrar o resultado.