

# FUNÇÕES



**INSTITUTO  
FEDERAL**  
Paraíba

Professor: Paulo de Tarso F. Júnior  
[paulodt@gmail.com](mailto:paulodt@gmail.com)

# Roteiro

- ▶ Introdução
- ▶ Módulos de Programas em C
- ▶ Biblioteca de Funções Matemáticas
- ▶ Funções
- ▶ Definições de Função
- ▶ Protótipos de Funções
- ▶ Arquivos de Cabeçalho
- ▶ Chamada de Funções por Valor e por Referência

# Introdução

- ▶ Divisão para a conquista

- ▶ Construção de programas a partir de partes ou componentes menores

- ▶ **Módulos**

- ▶ Maior facilidade de gestão de cada módulo do que do programa original
      - ▶ Componentes do programa que se repetem em pontos distintos

# Módulos de Programas em C

- ▶ Funções
  - ▶ Módulos em C
  - ▶ Possibilidade de combinação de funções definidas pelo usuário com funções das bibliotecas nos programas
    - ▶ Existência de uma vasta gama de funções na biblioteca padrão de C

# Módulos de Programas em C

- ▶ Chamadas de Funções
  - ▶ Invocação de funções
    - ▶ Explicitação do nome da função e passagem de argumentos (dados)
    - ▶ Realização de operações ou manipulações pela função
    - ▶ Retorno dos resultados pela função

# Módulos de Programas em C

- ▶ Chamadas de Funções

- ▶ Analogia

- ▶ Solicitação de execução de uma tarefa pelo patrão a um empregado
      - ▶ Aquisição de informações sobre a tarefa pelo empregado
      - ▶ Execução da tarefa
      - ▶ Retorno dos resultados
      - ▶ Ocultação da informação (patrão não conhece os detalhes)

# Bibliotecas de Funções

- ▶ Funções de Entrada e Saída
  - ▶ *#include <stdio.h>* - contém entrada *scanf()* e saída *printf()*
- ▶ Funções matemáticas
  - ▶ *#include <math.h>*
  - ▶ Execução de cálculos matemáticos comuns
- ▶ Processamento de textos
  - ▶ *#include <string.h>*
- ▶ Outras
  - ▶ *<conio.h>*, *<stdlib.h>*; *<time.h>*

# Biblioteca de Funções Matemáticas

- ▶ Formato para a chamada de funções
  - ▶ `printf( "%.2f", sqrt( 900.0 ) );`
    - ▶ Chamada da função *sqrt*, a qual retorna a raiz quadrada de seu argumento
    - ▶ Todas as funções matemáticas retornam dados do tipo ***double***
  - ▶ Argumentos → Constantes, variáveis ou expressões



# Funções

- ▶ Funções
  - ▶ Modularização de um programa
  - ▶ Todas as variáveis declaradas dentro de funções são variáveis locais
    - ▶ Conhecidas apenas no contexto da função
  - ▶ Parâmetros
    - ▶ Informação da comunicação entre funções
    - ▶ Variáveis locais

# Funções

- ▶ Benefícios de funções
  - ▶ Divisão para conquista
    - ▶ Desenvolvimento gerenciável de programas
  - ▶ Reusabilidade de *Software*
    - ▶ Uso de funções existentes como blocos para a construção de novos programas
    - ▶ Abstração
      - ▶ Ocultação de detalhes internos (funções da biblioteca)
  - ▶ Evita a repetição de código

# Definições de Funções

- Formato de Definição de uma Função

*Tipo\_do\_valor\_de\_retorno nome\_da\_função (lista de parâmetros )*

```
{  
    declarações e atribuições  
}
```

- *Nome\_da\_função* → Qualquer identificador válido

# Definições de Funções

- ▶ Formato de Definição de uma Função
  - ▶ *Lista\_de\_Parâmetros* → *Declaração de uma série de parametros* (= pares *tipo identificador*)
    - ▶ Um tipo deve ser listado explicitamente para cada parâmetro, caso contrário o parâmetro será considerado do tipo *int*

# Definições de Funções

## ► Formato de Definição de uma Função

*Tipo\_do\_valor\_de\_retorno nome\_da\_função (lista de parâmetros )*

*{*

*declarações e atribuições*

*}*

## ► Declarações e atribuições → Corpo da função (bloco de código)

- Variáveis podem ser declaradas dentro dos blocos (podem ser aninhadas)
- Funções não podem ser definidas dentro de outras funções

# Definições de Funções

- ▶ Formato de Definição de uma Função

- ▶ Retorno do Controle

- ▶ Quando não há retorno

- ▶ *return;*

- ▶ Se algo for retornado

- ▶ *return expression;*

# Definições de Funções

```
01  /* Determinação do máximo de três inteiros */
02  #include <stdio.h>
03
04  int maximo( int, int, int ); /* protótipo da função */
05  int main()
06  {
07      int a, b, c;
08
09      printf( "Digite três inteiros: " );
10      scanf( "%d%d%d", &a, &b, &c );
11      printf( "O maximo eh: %d\n", maximo( a, b, c ) );
12
13      return 0;
14  }
```

# Definições de Funções

```
15  /* Definição da função maximo */
16  int maximo( int x, int y, int z )
17  {
18      int max = x;
19
20      if ( y > max )
21          max = y;
22
23      if ( z > max )
24          max = z;
25
26      return max;
27  }
```

***Digite três inteiros: 22 85 17  
Maximo eh: 85***



# Protótipos de Funções

- ▶ Protótipo de uma Função
  - ▶ Nome da função
  - ▶ Parâmetros → O QUE a função recebe
  - ▶ Tipo de Retorno → Tipo de dado que a função retorna (*default* int)
  - ▶ Uso no processo de validação de funções
  - ▶ Necessidade de inclusão do protótipo apenas se a definição da função sucede a função principal
  - ▶ Função com o protótipo
    - ▶ `int maximo(int, int, int);`
      - ▶ Recebimento de **3** parâmetros *int*
      - ▶ Retorno de **1** dado *int*

# Protótipos de Funções

## ▶ Coerção de Argumentos

- ▶ Imposição de argumentos do tipo apropriado

- ▶ Exemplo

- ▶ Função `sqrt` → Possibilidade de chamada com um argumento `int`, embora o protótipo em *math.h* especifique um argumento *double*

- ▶ `printf("%.3f\n", sqrt(4));`

- ▶ Resultado gerado → Cálculo correto de `sqrt(4)` e impressão do valor 2.000

# Protótipos de Funções

- ▶ Regras de Promoção
  - ▶ Especificação de *como alguns tipos podem ser* convertidos para outros sem perda de dados
  - ▶ Possibilidade de cometimento de erros
    - ▶ Conversão de ***double*** em ***int*** → Truncamento da parte fracionária do valor ***double***
  - ▶ Aplicação automática a expressões contendo dois ou mais tipos de dados (mistas)

# Arquivos de Cabeçalho

- ▶ Arquivos de Cabeçalho
  - ▶ Contêm os protótipos das funções das bibliotecas referenciadas no programa
  - ▶ E.g. `<stdlib.h>` ,`<math.h>` ,`<conio.h>`
  - ▶ Necessidade de inclusão da(s) linha(s)
    - ▶ `#include <nome_do_arquivo>`
    - ▶ `#include <math.h>`

# Arquivos de Cabeçalho

- ▶ Arquivos-Cabeçalhos Customizados
  - ▶ Criação de arquivos com funções
  - ▶ Salvamento
    - ▶ <minha\_biblioteca.h>
  - ▶ Inclusão em outros arquivos
    - ▶ #include “minha\_biblioteca.h”
  - ▶ Reuso das funções

# Chamada de Funções por Valor e por Referência

- ▶ Uso na invocação de funções
- ▶ **Chamada por valor**
  - ▶ Cópia do(s) argumento(s) passado(s) para a função
  - ▶ Alterações do(s) argumento(s) na função não exercem influência sobre o(s) valor(es) original(ais)
  - ▶ Uso quando não há necessidade de alteração do argumento pela função
    - ▶ Prevenção contra alterações acidentais

# Chamada de Funções por Valor e por Referência

## ► Chamada por referência

- Passagem do(s) argumento(s) original(ais)
- Alterações do(s) argumento(s) na função implicam alterações no(s) original(ais)
- Uso apenas com funções confiáveis que precisem modificar a variável original
- Foco atual → Chamada por valor

# FUNÇÕES



**INSTITUTO  
FEDERAL**  
Paraíba

Professor: Paulo de Tarso F. Júnior  
[paulodt@gmail.com](mailto:paulodt@gmail.com)