

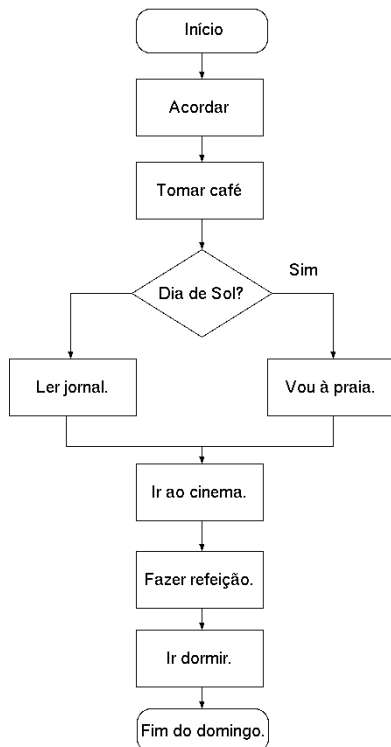
Ciência da Computação

Prof. Dr. Leandro Alves Neves

Aula 14

Algoritmos & Programação

Fluxograma para um domingo



Sumário

- Função
 - Protótipo
 - Escopo de Variáveis (Global e Local)
 - Passagem de Parâmetro (Valor)

1. Função: Definição

■ Definição

- ❑ Instruções organizadas em um bloco
 - Estruturação e Reutilização
- ❑ Visa resolver um problema específico
- ❑ **Pode retornar um valor**
 - Se não há retorno, então o tipo da função será **void**

■ Exemplos

- ❑ **printf()**: função que escreve na tela
- ❑ **scanf()**: função que permite entrada via teclado

2. Função: Protótipo

■ Protótipo:

```
tipo_retornado nome_função (<argumentos definem  
aridade>);
```

```
tipo_retornado nome_função (<argumentos>)  
{  
    Instruções;  
    <return tipo de argumento>;  
}
```

2. Função: Protótipo

```
#include <stdio.h>
#include <stdlib.h>
//Protótipo
```

Aridade 2

Tipo

```
int soma(int n, int n1);
```

```
int main()
{
```

Argumentos e Tipos

```
    int num_1, num_2;
```

```
    //Entrada dos dados
```

```
    printf("\nDigite o primeiro número.: ");
```

```
    scanf("%d",&num_1);
```

```
    printf("\nDigite o segundo número.: ");
```

```
    scanf("%d",&num_2);
```

Chamada

```
    printf ("\n A soma dos números é: %d", soma(num_1, num_2));
```

```
    printf("\n");
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

```
int soma(int n, int n1)
```

```
{
```

```
    return (n + n1);
```

```
}
```

Instruções

2. Função: Protótipo

```
#include <stdio.h>
#include <stdlib.h>
int soma(int n, int n1)
{
    return (n+n1);
}
```

■ Exemplo 2

```
int main()
{
    int num_1, num_2;

    //Entrada dos dados
    printf("\nDigite o primeiro número.: ");
    scanf("%d",&num_1);
    printf("\nDigite o segundo número.: ");
    scanf("%d",&num_2);

    printf ("\n A soma dos números é: %d", soma(num_1, num_2));
    printf("\n");
    system("PAUSE");
    return 0;
}
```

3. Função: Escopo de Variável

■ Escopo de Variáveis:

■ Variável Global

- Acessível em todos os escopos de um programa

- **Prefira variável local:** facilita organização, interpretação e permite otimizar o uso de recursos computacionais

■ Variável Local

- Acessível apenas no escopo em que foi declarada

3. Função: Escopo de Variável

```
#include <stdio.h>
#include <stdlib.h>
int soma(int n, int n1)
```

■ Variáveis locais

```
{
    return (n+n1);
}
```

n, n1 estão disponíveis apenas em soma

```
int main()
{
```

```
    int num_1, num_2;
```

■ Variáveis locais

```
    //Entrada dos dados
```

```
    printf("\nDigite o primeiro número.: ");
```

```
    scanf("%d", &num_1);
```

```
    printf("\nDigite o segundo número.: ");
```

```
    scanf("%d", &num_2);
```

```
    printf ("\n A soma dos números é: %d", soma(num_1, num_2));
```

```
    printf("\n");
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```


3. Função: Escopo de Variável

```
#include <stdio.h>
#include <stdlib.h>
//Escopo - Variáveis Globais
```

```
int num_1, num_2;
```

■ Variáveis Globais

```
int soma()
{
    return (num_1+num_2);
}
```

num_1, num_2 estão disponíveis em soma

```
int main()
{
    //Entrada dos dados
    printf("\nDigite o primeiro número.: ");
    scanf("%d",&num_1);
    printf("\nDigite o segundo número.: ");
    scanf("%d",&num_2);
```

num_1, num_2 estão disponíveis em main

```
printf ("\n A soma dos números %d e %d é: %d", num_1, num_2, soma());
printf("\n");
system("PAUSE");
return 0;
}
```

3. Função: Passagem de Parâmetro

- Passagem de Parâmetros:

- Valor

- Apenas o conteúdo da variável é acessado

- Não é possível realizar qualquer Alteração

- Referência

- O endereço de memória torna-se disponível (Uso do operador &)

- É possível realizar Alteração

4. Função: Passagem de Parâmetro

```
#include <stdio.h>
#include <stdlib.h>
int soma(int n, int n1)
```

■ Parâmetro por Valor

```
{
    return (n+n1);
}
```

n, n1 estão disponíveis apenas em *soma*

```
int main()
{
```

```
    int num_1, num_2;
```

```
    //Entrada dos dados
```

```
    printf("\nDigite o primeiro número.: ");
```

```
    scanf("%d",&num_1);
```

```
    printf("\nDigite o segundo número.: ");
```

```
    scanf("%d",&num_2);
```

```
    printf ("\n A soma dos números é: %d", soma(num_1, num_2));
```

```
    printf("\n");
```

```
    system("PAUSE");
```

```
    return 0;
```

```
}
```

Os valores das variáveis locais *num_1* e *num_2* **são associados** aos endereços das variáveis locais *n, n1*

4. Função: Passagem de Parâmetro

■ Parâmetro por Referência

```
#include <stdio.h>
#include <stdlib.h>
int soma(int n, int n1)
{
    return (n + n1);
}
```

```
void leitura (int *n, int *n1)
```

```
{
    //Entrada de Dados.
    //Manipulação das variáveis, via ponteiro
    printf("\nDigite o primeiro número.: ");
    scanf("%d",&*n);
    printf("\nDigite o segundo número.: ");
    scanf("%d",&*n1);
}
```

```
int main()
```

```
{
    int num_1, num_2;
    //Chamada com passagem de parâmetros por referência.
    leitura (&num_1, &num_2);
    printf ("\n A soma dos números é: %d", soma(num_1, num_2));
    printf("\n");
    system("PAUSE");
    return 0;
}
```

Ponteiros *n* e *n1* para acessar endereços de *num_1* e *num_2*

Os endereços de *num_1* e *num_2* são **associados** aos ponteiros locais *n*, *n1*

Exercícios

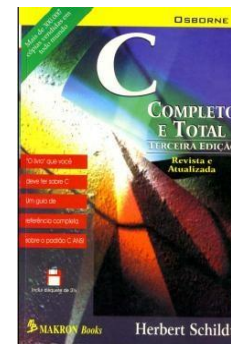
- Construa programas usando funções:
 1. Construa um programa com uma função para trocar os valores de dois números inteiros, ambos **passados como referência**. Os valores trocados devem ser apresentados na função principal.
 2. Faça um programa para ler um número e calcular o fatorial. O fatorial deve ser calculado em uma função, que recebe o número digitado por valor. O resultado deve ser apresentado na função principal.

Desafio

3. Desafio: Considere uma matriz com dimensões $N \times 4$ que armazena valores correspondentes a 3 notas de provas de cada um dos N alunos de uma turma. A matriz tem ainda um espaço para o armazenamento, na quarta coluna, para a média de cada um dos alunos nas três avaliações. Implemente as seguintes funções:
- lernotas: função do tipo void que recebe como parâmetros uma matriz $N \times 4$ e o valor de N considerado. A matriz deve ser recebida como um ponteiro. Esta função deverá ler e armazenar, nas 3 primeiras colunas da matriz, as 3 notas de provas de cada um dos N alunos da turma.
 - mediageral: função que retorna como resultado a média geral de uma turma (média das médias dos alunos da turma). Esta função receberá como parâmetros uma matriz $N \times 4$ e o valor de N considerado. Além disso, sua função deverá atualizar a 4ª coluna da matriz para incluir a média de cada um dos alunos nas três avaliações.
 - mostrarmedias: esta função deverá receber como parâmetros uma matriz $N \times 4$ e o valor de N considerado. Esta função deverá exibir os valores armazenados na 4ª coluna da matriz considerada.
 - Além disso, faça um programa principal que utilize as funções implementadas, na seguinte ordem: lernotas, mediageral e mostrarmedias.

Bibliografia

1. Schildt, H. C completo e total. São Paulo: Makron Books, 3ª ed, 1997.



2. Ascencio, A. F. G.; Campos, E. A. V. Fundamentos da Programação de Computadores: Algoritmos, Pascal, C/C++ e Java. 3ª. Ed. São Paulo: Pearson Education do Brasil, 2012.

