

SUBPROGRAMAÇÃO EM C

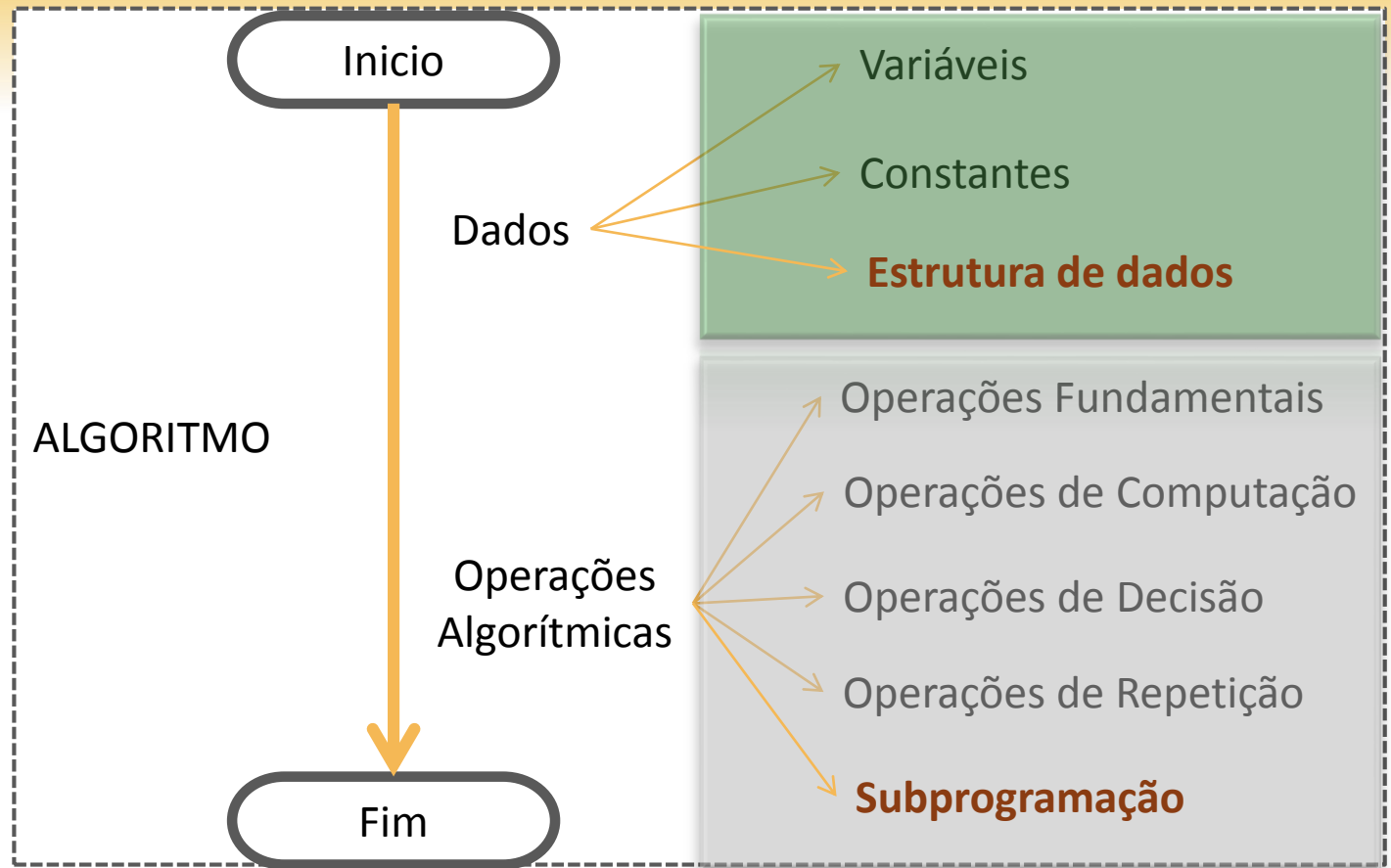
PASSAGEM DE PARÂMETROS

Prof. Josildo Silva

AGENDA

- ⦿ Noções de ponteiros
- ⦿ Lista de parâmetros
- ⦿ Passagem de parâmetros por valor
- ⦿ Passagem de parâmetros por referência
- ⦿ Exemplos

ALGORITMO COMPUTACIONAL



Conjunto ordenado de passos automatizáveis atuando sobre dados que definem um processo finito.

NOÇÕES DE PONTEIRO

VARIÁVEL

ENDEREÇO	CONTEÚDO
<code>int x;</code>	
X	
0x00001CFB	??
<code>x = 10;</code>	
X	
0x00001CFB	10

NOÇÕES DE PONTEIRO

- ⦿ Ponteiros são variáveis especiais que armazenam endereço de memória (endereço de variáveis)
- ⦿ Um ponteiro é um mecanismo de acesso indireto ao conteúdo de variáveis.
- ⦿ Acesso indireto significa sem utilizar o nome variável

NOÇÕES DE PONTEIRO

- ⊙ Uma variável tipo ponteiro permite manipular conteúdos de variáveis indiretamente através de dois operadores:

- ⊙ `*`

- acesso ao conteúdo de um endereço de memória

- ⊙ `&`

- acesso ao endereço de memória

NOÇÕES DE PONTEIRO

```
int x;  
int *y;
```

```
x = 10;
```

X	ENDEREÇO	CONTEÚDO
	0x00001CFB	10

Y	ENDEREÇO	CONTEÚDO
	0x00001A2D	???

NOÇÕES DE PONTEIRO

```
int x;  
int *y;
```

```
x = 10;
```

X	ENDEREÇO	CONTEÚDO
	0x00001CFB	10

```
y = &x;
```

Y	ENDEREÇO	CONTEÚDO
	0x00001A2D	???

NOÇÕES DE PONTEIRO

```
int x;  
int *y;
```

```
x = 10;
```

X	ENDEREÇO	CONTEÚDO
	0x00001CFB	10

```
y = &x;
```

Y	ENDEREÇO	CONTEÚDO
	0x00001A2D	0x00001CFB

NOÇÕES DE PONTEIRO

```
int x;  
int *y;
```

```
x = 10;
```

X	ENDEREÇO	CONTEÚDO
	0x00001CFB	10

```
y = &x;
```

Y	ENDEREÇO	CONTEÚDO
	0x00001A2D	0x00001CFB

```
*y = 20;
```

NOÇÕES DE PONTEIRO

```
int x;  
int *y;
```

```
x = 10;
```

X	ENDEREÇO	CONTEÚDO
	0x00001CFB	20

```
y = &x;
```

Y	ENDEREÇO	CONTEÚDO
	0x00001A2D	0x00001CFB

```
*y = 20;
```

NOÇÕES DE PONTEIRO

- ⦿ Declaração de ponteiro

- ⦿ `tipo_dado *nome_ponteiro;`

- ⦿ Inicialização de ponteiros

- ⦿ `nome_ponteiro = &nome_var;`

- ⦿ Modificação de conteúdo via ponteiro

- ⦿ `*nome_ponteiro = conteúdo;`

- ⦿ Neste caso o conteúdo pode ser uma constante ou uma variável.

LISTA DE PARÂMETROS

```
tipo_retorno Nome_Função(lista_parâmetros)
{
    sequencia de instruções;
}
```

Lista de parâmetros

- ✓ É o mecanismo para comunicação de dados entre subprogramas.

LISTA DE PARÂMETROS

- ⦿ **Conjunto de variáveis especiais para troca e/ou modificação de dados entre subprogramas.**
- ⦿ Como ocorre para todas as variáveis, a lista de parâmetros:
 - ⦿ Declaração: definição da lista de variáveis
 - ⦿ Atribuição: na ativação do subprograma quando os conteúdos são preenchidos.

LISTA DE PARÂMETROS

DEFINIÇÃO

- ⊙ A lista de parâmetros é definida ao mesmo tempo que se define o subprograma.
- ⊙ A definição da lista de parâmetros consiste em declarar o conjunto de variáveis que servirão para viabilizar troca/modificação de dados entre os subprogramas.

LISTA DE PARÂMETROS

DEFINIÇÃO

```
void subprog1 (int a, float b) ;  
void subprog2 (char c, float *d) ;
```

A lista de parâmetros:

- ✓ subprog1 é constituída de duas variáveis: variável “a” tipo numérico inteiro e a variável “b” tipo numérico real;
- ✓ subprog2 é constituída de duas variáveis: variável “c” tipo caractere e a variável “d” tipo ponteiro de float;

LISTA DE PARÂMETROS ATRIBUIÇÃO

```
void subprog1 (int a, float b) ;
```

A atribuição dos valores da lista de parâmetros ocorre no momento em que subprograma é acionado por outro subprograma:

```
subprog1 (2, 5) ;
```

Nesse momento as variáveis (parâmetros) “a” e “b” recebem os valores “2” e “5”, respectivamente.

LISTA DE PARÂMETROS ATRIBUIÇÃO

```
void subprog1(int a, float b);
```

```
int x=10, y=50;  
subprog1(x, y);
```

Nesse momento as variáveis (parâmetros) “a” e “b” recebem os conteúdos de “x” e “y”, respectivamente.

ESCOPO DE VARIÁVEIS

```
int z; //Variável global  
  
int subProg1(int a, float b)  
{  
    int x; float v;  
}  
}
```

Local

Local

PASSAGEM DE PARÂMETROS

- ⦿ O processo de atribuir conteúdos às variáveis da lista de parâmetros é conhecido como passagem de parâmetros.
- ⦿ Os métodos para passagem de parâmetros são dois:
 - ⦿ Passagem por valor
 - ⦿ Passagem por referência

PASSAGEM DE PARÂMETROS POR VALOR

- ⊙ As variáveis da lista de parâmetros recebem cópias de conteúdos dos argumentos correspondentes no momento da chamada do subprograma.
- ⊙ E qualquer mudança produzidas nas variáveis da lista de parâmetros por ação do subprograma não produzem mudanças nos argumentos originais.

PASSAGEM DE PARÂMETROS POR REFERÊNCIA

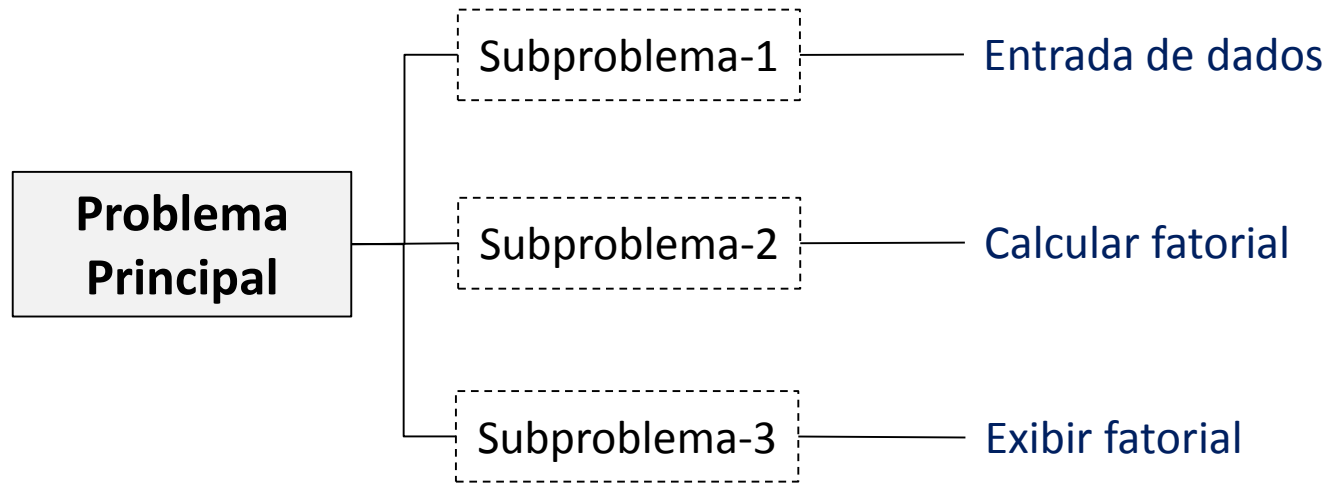
- ⊙ As variáveis da lista de parâmetros recebem cópias de endereços dos argumentos correspondentes no momento da chamada do subprograma.
- ⊙ E qualquer mudança produzida nos conteúdos das variáveis da lista de parâmetros por ação do subprograma produzem mudanças nos argumentos originais.

SUBPROGRAMAS EM C

DEFINIÇÃO E ATIVAÇÃO

Exemplo 1

Dado um número inteiro obrigatoriamente não negativo deve-se exibir seu fatorial.



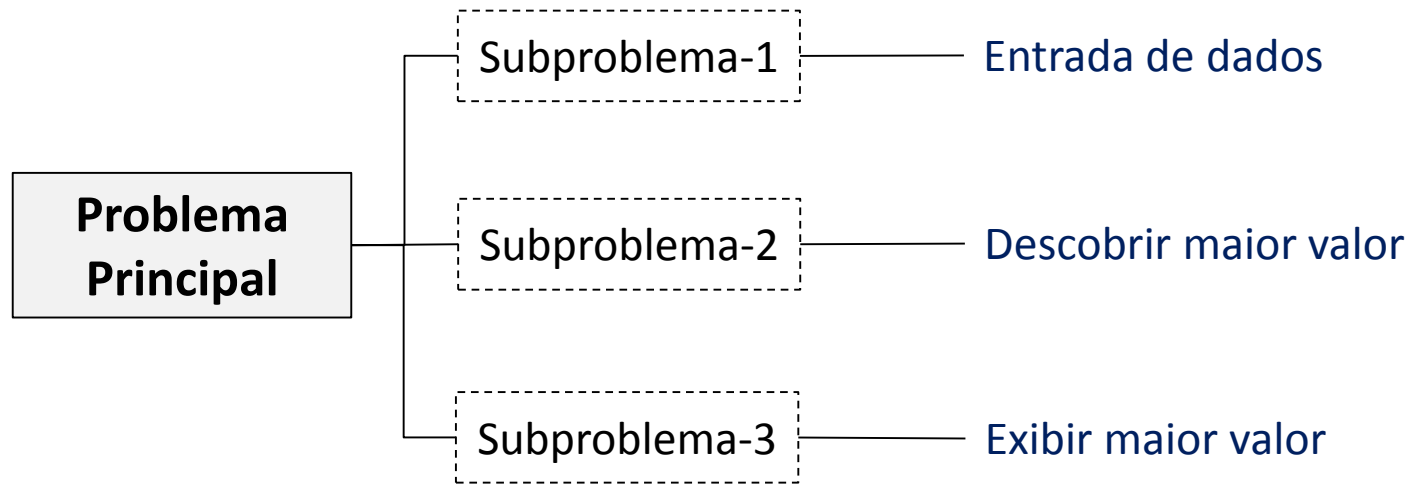
Todos os subproblemas são resolvidos com subprogramas!!

SUBPROGRAMAS EM C

DEFINIÇÃO E ATIVAÇÃO

Exemplo 2

Desenvolva um programa que receba um conjunto de salários e o informe o maior salário. A entrada de dados se encerra mediante questionamento ao usuário.



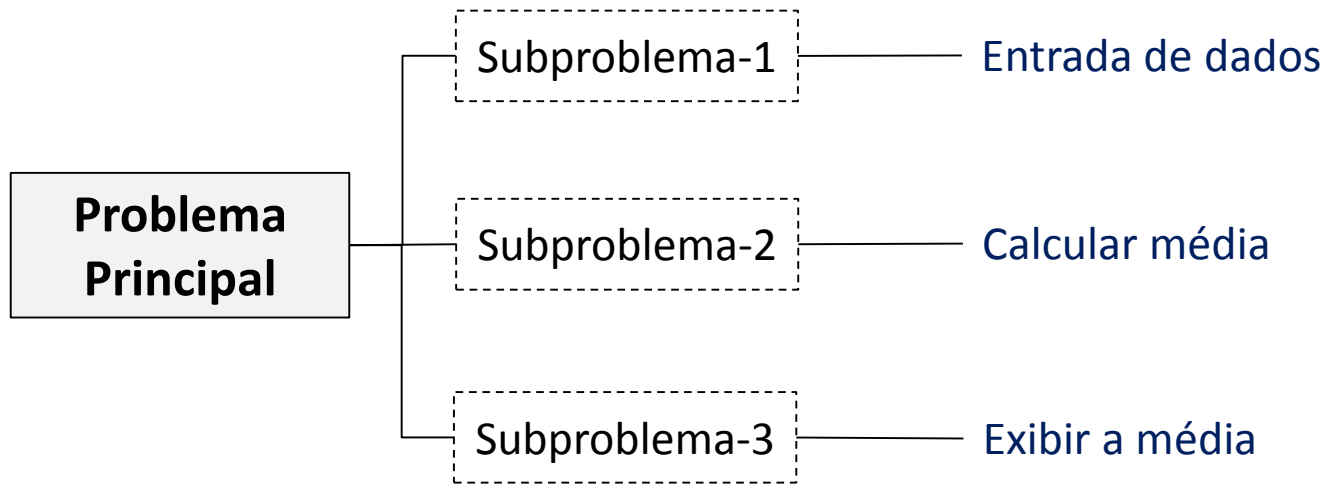
Todos os subproblemas são resolvidos com subprogramas!!

SUBPROGRAMAS EM C

DEFINIÇÃO E ATIVAÇÃO

Exemplo 3

Desenvolva um programa que receba um conjunto de salários e informe a média de salários. A entrada de dados se encerra mediante questionamento ao usuário.



Todos os subproblemas são resolvidos com subprogramas!!

REFERÊNCIAS

1. C Completo e Total. Herbert Schild. Person Universidades.
2. Linguagem C Completa e Descomplicada. André Backes. Elsevier, Rio de Janeiro.