



UNIVERSIDADE FEDERAL DO PIAUÍ – UFPI
Campus Senador Helvídio Nunes de Barros - CSHNB
Curso Bacharelado em Sistemas de Informação

ALGORITMOS E PROGRAMAÇÃO I

Glauber Dias Gonçalves
ggoncalves@ufpi.edu.br

CONTEÚDO

- Conceito de funções
- Codificação de funções na linguagem C
- Atividades Práticas

FUNÇÃO

- Módulo do algoritmo com uma tarefa específica
- Objetivo: modularizar ou organizar o algoritmo
 - Estratégia: dividir para conquistar
- Pode ou não retornar um valor
- Envolve dois passos:
 - **Protótipo**: no início do algoritmo
 - **Chamada**: dentro de alguma função do algoritmo
 - **Declaração**: trecho do algoritmo com especificação da função

PROTÓTIPO

tipo nome_da_função (lista_de_parâmetros)

- **tipo:** a informação retornada da função;
 - se não retornar nada, seu tipo deve ser void;
- **nome_da_função:** mesma regra para variáveis
- **parâmetros:** lista de tipos (e variáveis) que serão passados como argumentos para a função
 - pode ser vazio.

DECLARAÇÃO

tipo nome_da_função (lista_de_parâmetros)

início

declaração de variáveis

comandos

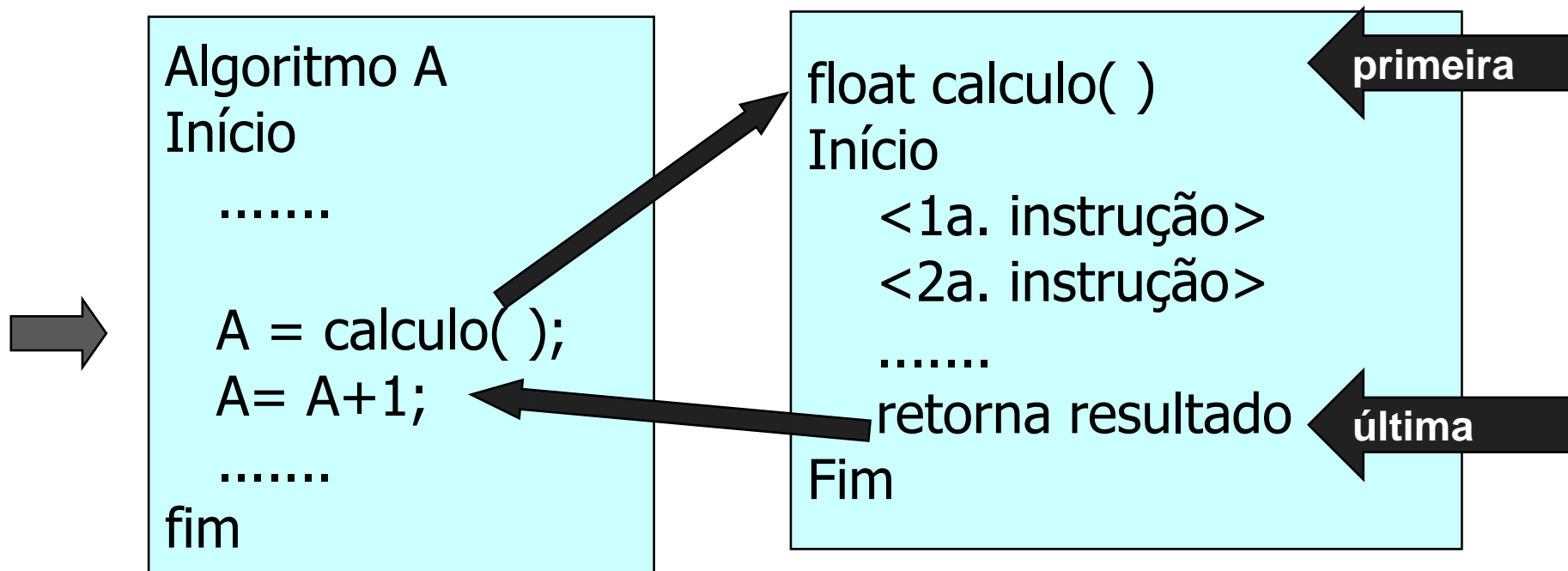
*retorno (expressão) /*opcional*/*

fim

- a primeira linha é idêntica ao protótipo
- o ***retorno*** serve para indicar o valor a ser retornado,
 - pode aparecer em qualquer ponto da função
 - pode aparecer em mais de um ponto.

CHAMADA

- Transfere controle para o função chamada
- Executa até o fim da função
- retorna o controle de volta para o local de chamada.



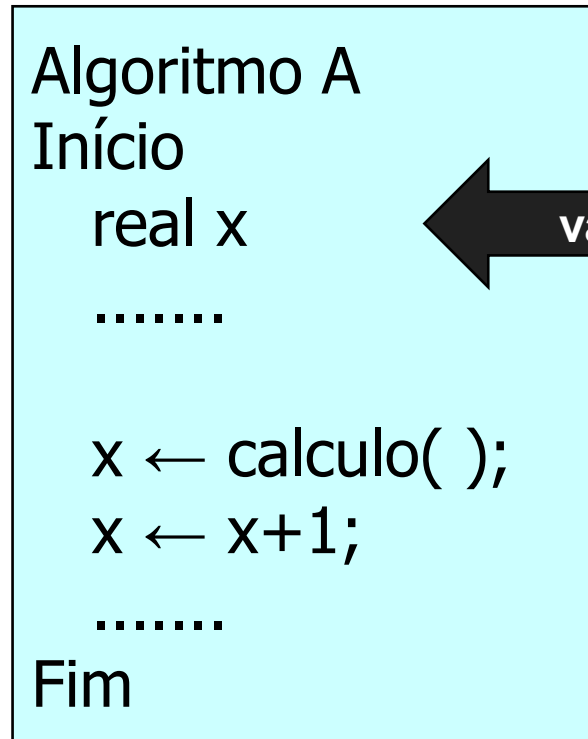
VARIÁVEIS E FUNÇÕES

- Uma função pode usar variáveis:
 - Locais: válidas apenas dentro da função onde foi declarada
 - Globais: válidas em todas as funções do algoritmo

VARIÁVEIS LOCAIS

- Espaço de memória é alocado no início da execução da função e liberado no final;
- Podem ser declaradas em qualquer parte do bloco que compõe a função;
- Só podem ser usadas pela função à qual pertencem;
- Valores são perdidos quando a função termina.

VARIÁVEIS LOCAIS



variável local

Válida apenas
dentro do
Algoritmo A

VARIÁVEIS GLOBAIS

- Declaradas fora de funções;
- Podem ser usadas em qualquer função;
- Devem ficar antes da declaração da função principal

VARIÁVEIS GLOBAIS

```
float resultado
```

Algoritmo A

Início

.....

```
resultado = 0
```

```
calculo( );
```

```
escreve(resultado)
```

.....

fim

```
inteiro calculo( )
```

Início

```
resultado *= 10
```

```
retorna 0
```

Fim

RETORNO DA FUNÇÃO

- Formas de utilização
 - Sem retorno de valor;
 - Com retorno de valor.
- Exemplo
 - Fatorial na linguagem C

FATORIAL (SEM RETORNO DE VALOR)

inteiro n

inteiro fatorial ()

Programa testaFatorial

Início

 leia(n)

 fatorial ()

 retorna 0

Fim

inteiro fatorial ()

 inteiro i, fat = 1

 para i de 1 até n faça

 fat *= i

 fim_para

 escreva(fat)

 retorna 0

fim

COMANDO RETORNO

- Retorna um resultado para o local de chamada da função
 - pode ser armazenado em uma variável
 - pode ser usado em algum comando ou expressão
- Finaliza a função no local do retorno
- Pode aparecer mais de uma vez na função
 - apenas um será executado a cada chamada da função

FATORIAL (COM RETORNO DE VALOR)

inteiro n

inteiro fatorial ()

Programa testaFatorial

 inteiro n, fat

Início

 leia(n)

 fat ← fatorial ()

 escreva(fat)

 retorna 0

Fim

inteiro fatorial ()

 inteiro i, fat = 1

 se n = 0 então

 retorna 0

 para i de 1 até n faça

 fat *= i

 fim_para

 retorna fat

fim

Codificação na Linguagem C

FATORIAL (SEM RETORNO DE VALOR)

```
#include <stdio.h>

int n; //Variavel global
void fatorial(void); //Prototipo da função fatorial

int main(){
    printf("Digite o valor de n: ");
    scanf("%d",&n);

    fatorial(); //Chamada da funcao fatorial

    return 0;
} //fim da funcao main

//Criacao da funcao fatorial
void fatorial(){
    int i, fat = 1;
    for(i=1; i<=n; i++){ //A funcao fatorial utiliza o valor de n
        fat *= i;
    } //fim do for
    printf("Fatorial = %d\n",fat);
} //fim da funcao fatorial
```

FATORIAL (COM RETORNO DE VALOR)

```
#include <stdio.h>

int n; //Variavel global
int fatorial(void); //Prototipo da função fatorial

int main(){
    int resultado; //Variavel que recebe o resultado
    printf("Digite o valor de n: ");
    scanf("%d",&n);

    resultado = fatorial(); //Chamada da funcao fatorial

    printf("Resultado = %d",resultado);
    return 0;
} //fim da funcao main

//Criacao da funcao fatorial
int fatorial(){
    int i, fat = 1;
    for(i=1; i<=n; i++){ //A funcao fatorial utiliza o valor de n
        fat *= i;
    } //fim do for

    return fat; //Retornando o valor da variavel fat
} //fim da funcao fatorial
```

PRÁTICA 1

- Modificar o código da função fatorial em C com retorno de valor para receber o valor n via passagem de parâmetro.
 - Fazer rascunho no slide seguinte e testar!

PRÁTICA 1

```
#include <stdio.h>

int n; //Variavel global
int fatorial(void); //Prototipo da função fatorial

int main(){
    int resultado; //Variavel que recebe o resultado
    printf("Digite o valor de n: ");
    scanf("%d",&n);

    resultado = fatorial(); //Chamada da funcao fatorial

    printf("Resultado = %d",resultado);
    return 0;
} //fim da funcao main

//Criacao da funcao fatorial
int fatorial(){
    int i, fat = 1;
    for(i=1; i<=n; i++){ //A funcao fatorial utiliza o valor de n
        fat *= i;
    } //fim do for

    return fat; //Retornando o valor da variavel fat
} //fim da funcao fatorial
```

PRÁTICA 2

- Fazer uma função para calcular raízes x' e x'' de uma equação quadrática com o seguinte protótipo:

void calculaRaizes(float a, float b, float c);

$$\Delta = b^2 - 4.a.c$$

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

PRÁTICA 3

- Mostre a diferença da função *calculaRaizes* com o uso de variáveis locais e variáveis globais:

void calculaRaizes(float a, float b, float c);

$$\Delta = b^2 - 4.a.c$$

$$x = \frac{-b \pm \sqrt{\Delta}}{2a}$$

SUMÁRIO

- Funções:
 - Recurso para modularizar ou organizar o algoritmo
 - Estratégia: dividir para conquistar
 - Pode ou não retornar um valor
- Envolve três passos:
 - **Protótipo**: no início do algoritmo
 - **Chamada**: dentro de alguma função do algoritmo
 - **Declaração**: trecho do algoritmo com especificação da função
- Lista de atividades 08: questões 06-10