

# **INF01202**

## **Algoritmos e Programação**

### **Modalidade Ead – Turma H**

**Material de apoio: capacitar a programação  
usando funções**

# FUNÇÕES com retorno

# Retorno de valores nos nomes das funções: comando **return**

Algumas funções já utilizadas anteriormente, como **sqrt**, **pow**, foram usadas em expressões.

Ex.:

```
raiz1 = (-b + sqrt(det))/(2*a);
```

```
elevado_ao_cubo = pow(num, 3);
```

Isso foi/é possível porque as funções podem retornar um valor em seu nome.

Essa possibilidade pode ou não ser explorada ao desenvolver-se uma função.

As funções de tipo **void**, não pré-definidas, que foram desenvolvidas até o momento, não exploram essa possibilidade. São ativadas como se fossem comandos e não ocorrem dentro de expressões.

## Retorno de valores nos nomes das funções: comando **return** (cont.)

Se uma função é declarada com tipo diferente de **void** (**int**, **char**, **float**, etc.) significa que ela pretende explorar a possibilidade de retorno de um valor em seu nome e então pode ser usada em expressões.

Lembrar que funções sem tipo declarado são consideradas pelo sistema como sendo funções inteiras.

Uma função que retorna um valor em seu nome deve conter pelo menos uma ocorrência do comando **return**, uma vez que é pela execução de um comando **return** que um valor é atribuído ao nome de uma função.

## Exemplo de função inteira

## Seja o programa **determinamaior**, que apresenta o maior de dois valores inteiros

```
//determina o maior de dois valores inteiros
#include <stdio.h>
#include <stdlib.h>
int acha_maior(int num1, int num2);
int main ( )
{
    system("color 70");
    int numero1, numero2;
    printf("\nForneca o primeiro valor: ");
    scanf("%d", &numero1);
    printf("\nForneca o segundo valor: ");
    scanf("%d", &numero2);
    printf("\nEntre os valores %d e %d, o maior eh %d",
           numero1, numero2, acha_maior(numero1, numero2));
    printf("\n");
    system("pause");
    return 0;
}
```

```
int acha_maior(int num1, int num2)
{
    if (num1 > num2)
        return num1;
    else
        return num2;
}
```

protótipo da função  
**acha\_maior**.

Outra versão possível  
deste protótipo:

**int acha\_maior(int, int);**

Função **acha\_maior** é uma  
função inteira: retorna  
um valor inteiro em seu  
nome.

## Exemplo de função inteira

Execução do programa **determinamaior**,  
que apresenta o maior de dois valores inteiros

 G:\EAD2007\FUNCOES\determinamaior.exe

Forneca o primeiro valor: 56

Forneca o segundo valor: 89

Entre os valores 56 e 89, o maior eh 89

Pressione qualquer tecla para continuar. . .

## Exemplo de função tipo *char*

Seja o programa  
**apresconc**,  
que apresenta o  
conceito de alunos a  
partir de suas notas

protótipo da função  
**devolveconc**.

Outra versão possível  
deste protótipo:

**char** **vevolveconc**(int);

Função **devolveconc** é  
uma função tipo **char**:  
retorna um caractere em  
seu nome.

//apresenta o conceito de n alunos

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAXNOTA 10
#define MAXALUNOS 5
char devolveconc (int nota);
int main ( )
{
    int i, notas[MAXALUNOS];
    printf("\nForneca as notas dos %d alunos\n", MAXALUNOS);
    for (i = 0; i < MAXALUNOS; i++)
    {
        do
        {
            printf("\nNota do aluno %d: ", i + 1 );
            scanf("%d", &notas[i]);
            if (notas[i] < 0 || notas[i] > MAXNOTA)
                printf("\nNota deve ser entre 0 e %d!!!\n", MAXNOTA);
        }
        while (notas[i] < 0 || notas[i] > MAXNOTA);
    }
    printf("\nAluno Nota Conceito");
    for(i = 0; i< MAXALUNOS; i++)
        printf("\n %2d    %2d    %2c\n",
            i + 1, notas[i], devolveconc(notas[i]));
    system("pause");
    return 0;
}
...
```

Seja o programa **apresconc**,  
que apresenta o conceito de alunos a partir de suas notas

Exemplo de  
função tipo *char*

```
char devolveconc (int nota)
```

```
{  
  switch (nota)  
  {  
    case 0:  
    case 1:  
    case 2:  
    case 3:  
    case 4:  
    case 5: return 'D';  
            break;  
    case 6:  
    case 7: return 'C';  
            break;  
    case 8:  
    case 9: return 'B';  
            break;  
    case 10: return 'A';  
  }  
}
```

Função **devolveconc** é  
uma função tipo **char**:  
retorna um caractere em  
seu nome.



# Execução do programa **apresconc**, que apresenta o conceito de alunos a partir de suas notas

 E:\EAD2007\FUNCOES\apresconc.exe

Forneca as notas dos 5 alunos

Nota do aluno 1: 3

Nota do aluno 2: 6

Nota do aluno 3: 7

Nota do aluno 4: 8

Nota do aluno 5: 10

Aluno	Nota	Conceito
-------	------	----------

1	3	D
---	---	---

2	6	C
---	---	---

3	7	C
---	---	---

4	8	B
---	---	---

5	10	A
---	----	---

Pressione qualquer tecla para continuar. . . \_

# Quando uma função encerra sua execução?

Uma função encerra sua execução quando:

- o fim do seu código é atingido;

ou

- um comando **return** é encontrado e executado.

Sobre comandos **return**:

vários **return** podem existir em uma mesma função, mas é quando um deles é ativado que a função se encerra (ver no exemplo anterior a função **acha\_maior**).

## APROVEITAMENTO DO VALOR RETORNADO POR UMA FUNÇÃO

Nem sempre se utiliza o valor retornado por uma função. A função *printf*, por exemplo, retorna um valor inteiro que é o número de caracteres apresentados. Regra geral esse valor é ignorado.

## Seja o programa **testaprintf** (gerado a partir do **determinamaior**)

```
//mostra o valor retornado por printf
#include <stdio.h>
#include <stdlib.h>
int acha_maior(int valor1, int valor2);
int main ( )
{
    system("color 70");
    int numero1, numero2, retorno_printf;
    printf("\nForneca o primeiro valor: ");
    scanf("%d", &numero1);
    printf("\nForneca o segundo valor: ");
    scanf("%d", &numero2);
    printf("\n123456789012345678901234567890123456789012345678901234567890\n");
    retorno_printf = printf("\nEntre os valores %d e %d, o maior eh %d",
        numero1, numero2, acha_maior(numero1, numero2));
    printf("\nRetorno do printf: %d\n", retorno_printf);
    printf("\n");
    system("pause");
    return 0;
}
int acha_maior(int num1, int num2)
{
    if (num1 > num2)
        return num1;
    else
        return num2;
}
```

## Execução do programa **testaprintf**,

 G:\EAD2007\FUNCOES\testaprintf.exe

Forneca o primeiro valor: 45

Forneca o segundo valor: 89

12345678901234567890123456789012345678901234567890

Entre os valores 45 e 89, o maior eh 89

Retorno do printf: 40

Pressione qualquer tecla para continuar. . .

# Observações importantes sobre funções em C

- ✓ Todo programa em C tem pelo menos uma função: a função **main**.
- ✓ Em uma função em C pode-se escrever qualquer comando da linguagem.
- ✓ Um parâmetro em uma função é uma variável local à função.
- ✓ O nomes das variáveis declaradas no cabeçalho de uma função são independentes dos nomes das variáveis usadas para chamar a função.
- ✓ O número e tipo dos parâmetros usados para chamar uma função deve ser igual ao número e tipo dos parâmetros declarados no cabeçalho da função.

## Observações importantes sobre funções em C (cont.)

- ✓ Qualquer tipo de dado pode ser tipo de um parâmetro de uma função.
- ✓ Qualquer expressão válida em C pode ser argumento para uma função.
- ✓ Ao ser ativada uma função, seus parâmetros (argumentos) são inicializados com os valores daquele momento dos argumentos usados para ativar a função.
- ✓ Uma função tipo **void** é chamada como se fosse um comando.
- ✓ Uma função com tipo não **void** tipicamente é chamada dentro de outro comando ou expressão, se for interessante usar seu valor de retorno.
- ✓ Em C não é possível definir-se uma função dentro de outra função. Mas é possível chamar-se uma função dentro de outra função.