

# **PROGRAMAÇÃO DE COMPUTADORES (PC)**

**ENGENHARIA DA COMPUTAÇÃO**

**Modularização Continuação**

**Profa. Dra. Lúcia F. A. Guimarães**



- **Funções em C**

- **Passagem por REFERÊNCIA**

PONTEIROS

- Transmite um **endereço de memória**, para o subprograma chamado, proporcionando, assim, acesso à célula que armazena o parâmetro real.

**Portanto o parâmetro real PODE ser alterado!!!**

- Na função: `Func(int *num)` → ponteiro para um inteiro de nome “num”.
- Dentro da função: o conteúdo armazenado pelo ponteiro → `*num`.
- Na chamada da função: `Func(&meuInteiro)` → referência do inteiro de nome “meuInteiro”



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

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```



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

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquematicamente, na memória



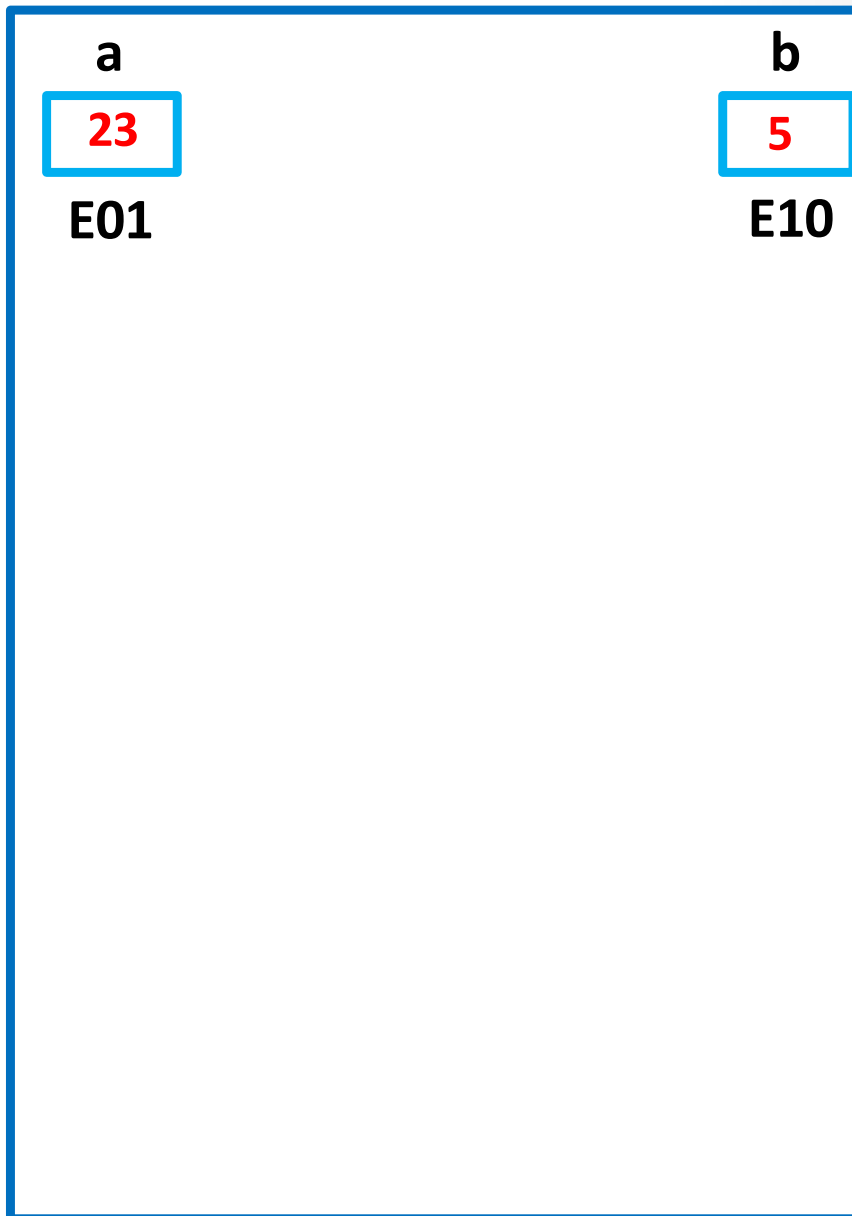


```
#include <stdio.h>
#include <stdlib.h>

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquemáticamente, na memória





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

```
void troca(int num, int *num1)
```

```
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}
```

```
main()
```

```
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquemáticamente, na memória

a

23

E01

b

5

E10

num

23

E50

num1

E10

E80

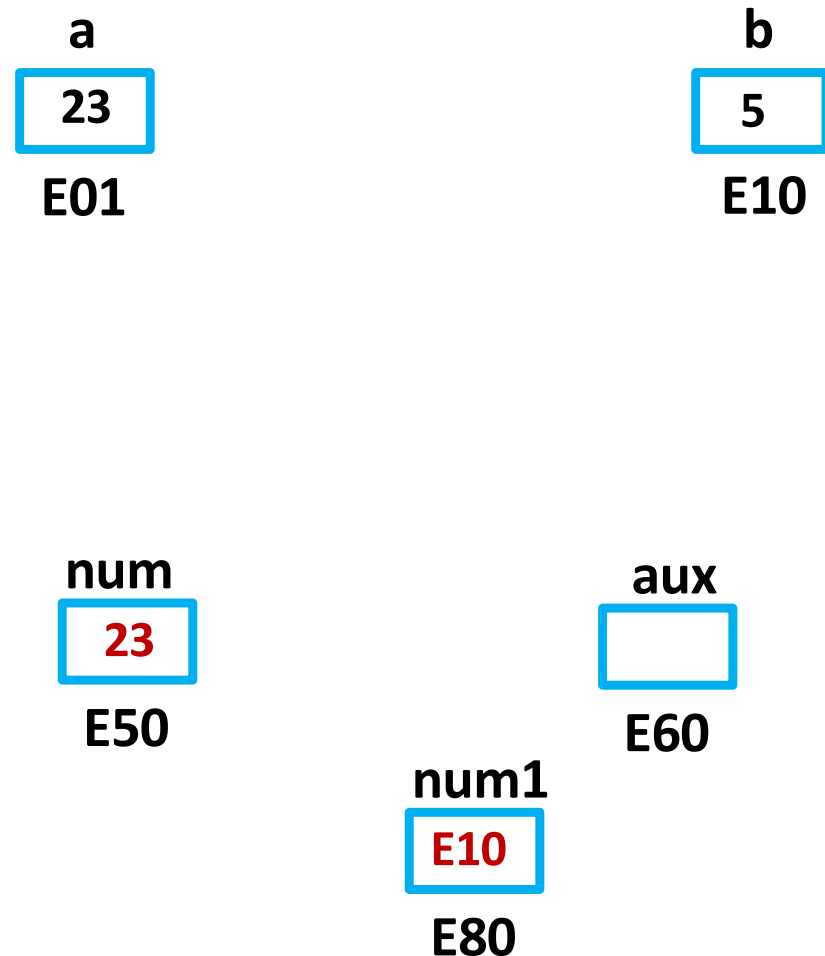


```
#include <stdio.h>
#include <stdlib.h>

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquemáticamente, na memória



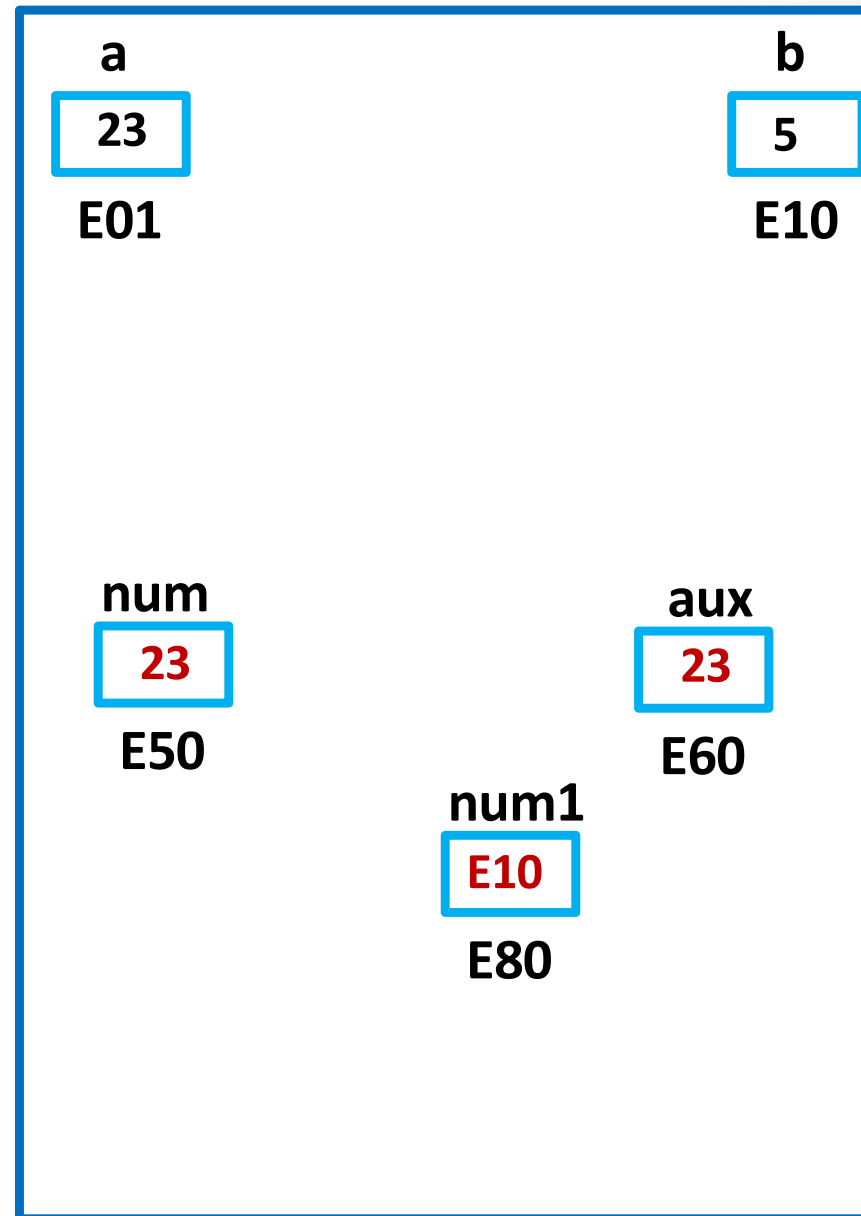


```
#include <stdio.h>
#include <stdlib.h>

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquemáticamente, na memória





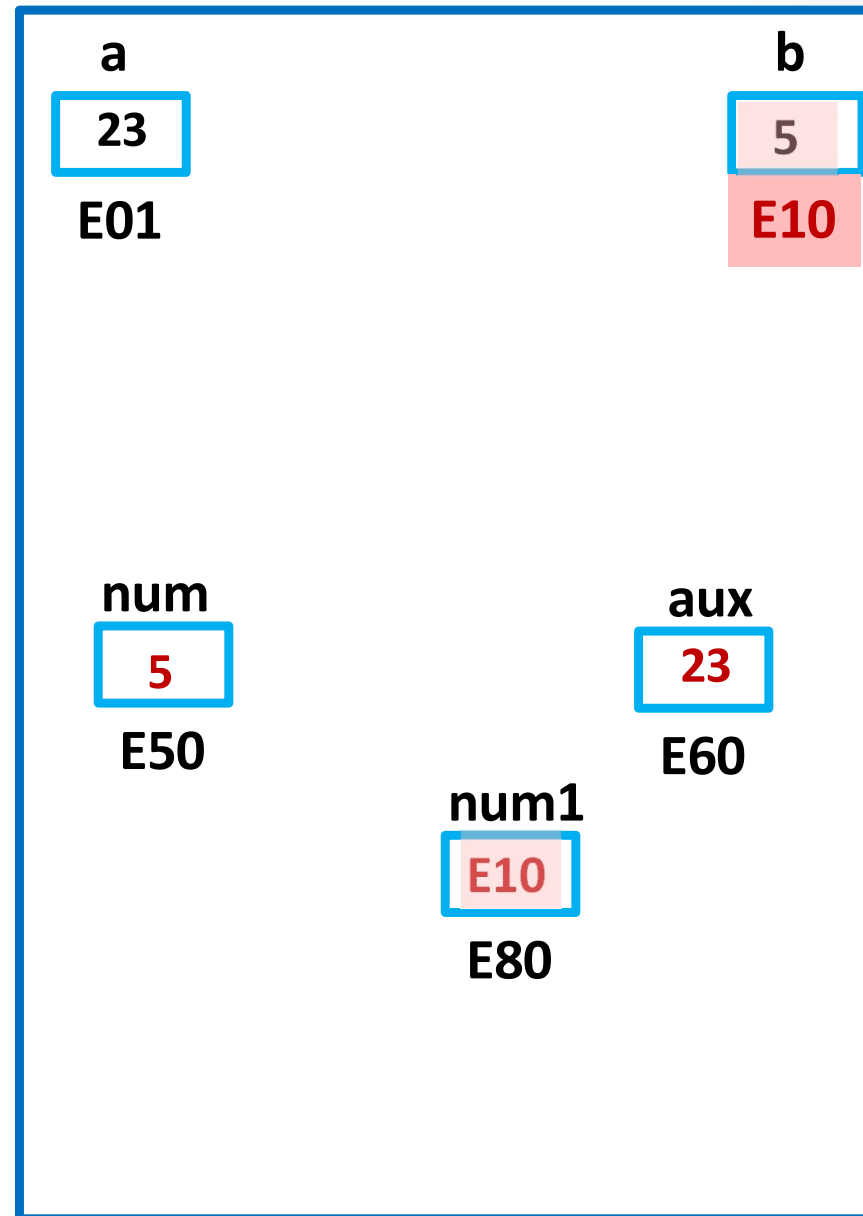


```
#include <stdio.h>
#include <stdlib.h>

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquemáticamente, na memória



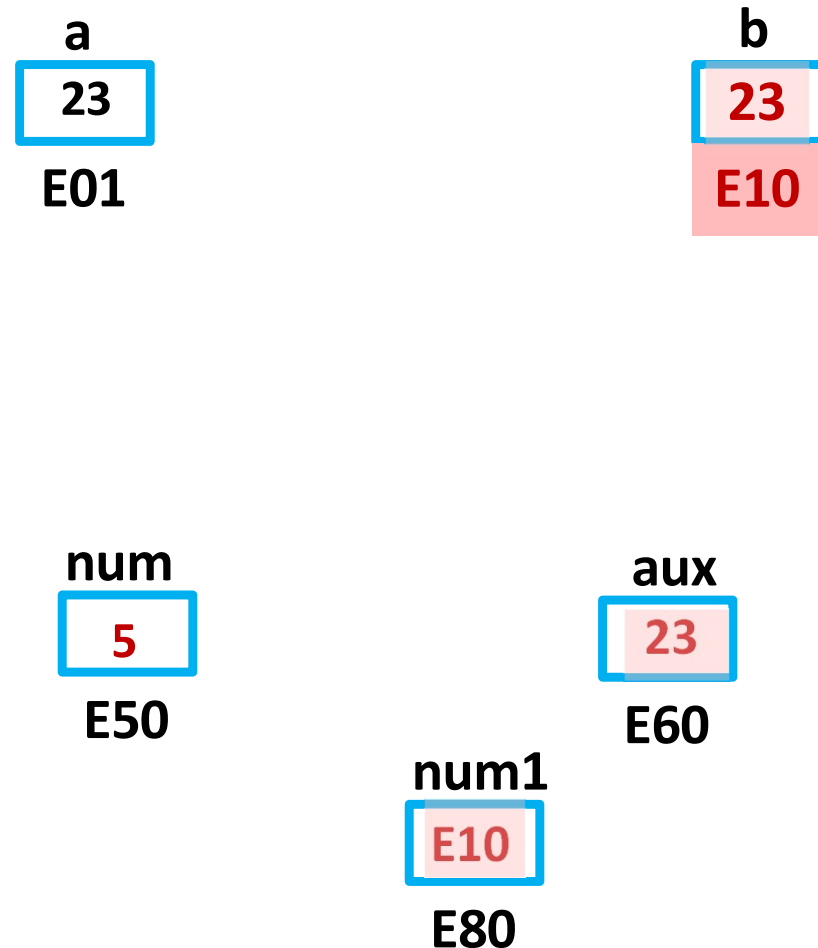


```
#include <stdio.h>
#include <stdlib.h>

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquematicamente, na memória





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

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquematicamente, na memória

a

23

E01

b

23

E10

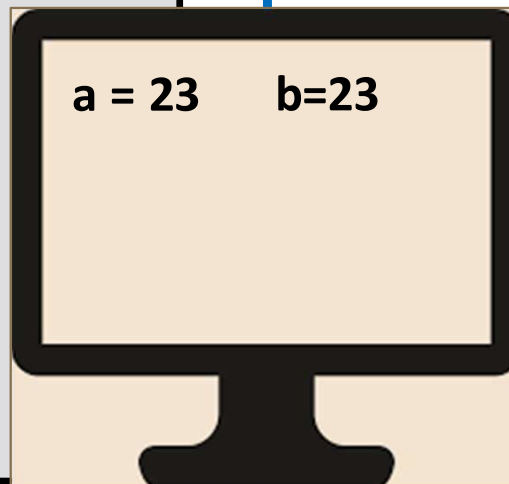


```
#include <stdio.h>
#include <stdlib.h>

void troca(int num, int *num1)
{
    int aux;
    aux=num;
    num=*num1;
    *num1=aux;
}

main()
{
    int a, b;
    a=23;
    b=5;
    troca(a,&b);
    printf("a=%d b=%d", a,b);
}
```

Esquematicamente, na memória





- **Passagem de Parâmetros**

- **Concluindo:**

- A passagem por **Valor NÃO Altera** o valor da variável no programa chamador.
- A passagem por **Referência PODE ALTERAR** o valor da variável no programa chamador.

- **Exercícios**

**1.** Elabore um programa que leia um número inteiro e construa os seguintes procedimentos

- **int soma\_digit(int a)** – este procedimento irá somar os dígitos desse número inteiro
- **void Maior\_Digit(int N, int \*M)** – este procedimento irá determinar o maior dígito desse número inteiro
- Os resultados deverão ser impressos no programa principal

**Atenção!!! Os cabeçalhos da função deverão ser respeitados, ou seja, você deve construir a função a partir desse cabeçalho**

## • Vamos Exercitar um pouco....

## Modularização

2. Construa um programa em C que leia um número inteiro não negativo e determine a soma dos seus divisores. A soma dever ser efetuada através de uma função **somadiv** e o resultado impresso no programa principal. O protótipo da função é:  
**void somadiv(int x, int\*y);**
3. Faça um programa que lê um número não determinado de valores inteiros e positivos, e para cada valor lido:
  - a) calcula e escreve o seu fatorial;
  - b) calcula e escreve os seus divisores;
  - c) Você deverá construir o programa, elaborando as seguintes funções:
    - **Ler\_num(int &x)** – função que lê um valor inteiro
    - **int Fatorial(int N)** – função que retorna o fatorial de um número inteiro N
    - **void divisores(int valor)** – função que imprime os divisores de um número