

Ex.: função troca

```
#include <stdio.h>

void troca(int i, int j){
    int t;
    t=i;
    i=j;
    j=t;
}

int main(){
    int a=5, b=10;
    printf("\na=%d, b=%d\n", a, b);
    troca(a, b); // passagem por valor
    printf("a=%d, b=%d\n", a, b);
    printf("\n");
}
```

essa função
faz a troca ???

```
a=5, b=10
a=5, b=10
```

Ex.: função troca

```
#include <stdio.h>

void troca(int i, int j){
    int t;
    t=i;
    i=j;
    j=t;
}

int main(){
    int a=5, b=10;
    printf("\na=%d, b=%d\n", a, b);
    troca(a, b); >> // passagem por valor
    printf("a=%d, b=%d\n", a, b);
    printf("\n");
}
```

essa função
faz a troca ???

```
a=5, b=10
a=5, b=10
```

```
#include <stdio.h>

void troca(int *i, int *j){
    int t;
    t=*i;
    *i=*j;
    *j=t;
}

int main(){
    int a=5, b=10;
    printf("\na=%d, b=%d\n", a, b);
    troca(&a, &b); >> // passagem por ref
    printf("a=%d, b=%d\n", a, b);
    printf("\n");
}
```

“i” passa a ser outro nome
p/ “a” e “j” passa a ser
outro nome p/ “b”

```
a=5, b=10
a=10, b=5
```

Passagem de parâmetros por referência

- Passa o endereço do valor
- Alterações nos parâmetros afetam o conteúdo da variável na função principal
- P/ alterar as variáveis passadas p/ uma função → declarar os parâmetros como ponteiros
 - ponteiro: "referência" necessária p/ alterar a variável
 - chamada da função: **&** precede as variáveis passadas por referência
 - parâmetros formais: ***** precede os parâmetros (declarados como ptr)

Exercícios: passagem por referência

- 1) Faça um programa que leia um número N e N valores inteiros. Para cada valor lido, chame uma função que calcule e retorne (na mesma variável) a quantidade de divisores do mesmo.
 - 2) Faça uma função que receba um número N como parâmetro por valor e 2 inteiros $p1$ e $p2$, por referência. A função deve retornar em $p1$ o maior número primo menor que N e em $p2$ o menor número primo maior que N .
 - 3) Faça uma função que receba um número unsigned int como parâmetro e o devolva escrito ao contrário (na mesma variável). Ex.: $431 \rightarrow 134$
 - 4) Faça uma função que receba um número inteiro n como parâmetro por valor e 2 inteiros x e y como parâmetros por referência. A função deve “devolver” em x o maior número primo menor que n e em y o menor número primo maior que n . Implemente também a função principal, responsável pela leitura e escrita dos valores manipulados na função. (obs.: os parâmetros são recebidos vazios e “voltam” com os valores calculados)
-

Strings: passagem de parâmetros

Sempre por referência (assim como em vetores)

O tamanho da string não precisa ser especificado na definição da função

- Strings de qualquer tamanho podem ser passadas na chamada da função

```
void func (char str1[]);
```

- Se uma string é passada como parâmetro, qualquer alteração afeta a string “original”

```
str = "abc";  
passaParaMaiuscula(str);  
printf("%s", str);           // vai imprimir "ABC"
```

Exercícios de string com função por referência

- 5) Faça um programa que leia uma string e chame uma função que inverta esta string. A string invertida deve ser impressa pela função main.
- 6) Faça um programa que leia uma string e chame uma função que remova todos os caracteres repetidos da string. A nova string deve ser impressa pela função main.
- 7) Faça uma função que receba uma string e duplique cada caractere dessa string. A string duplicada deve ser impressa pela função main.
- 8) Faça uma função que receba uma string e remova todos os caracteres repetidos da string. A função deve retornar quantidade de caracteres removidos para a função main, que deve imprimir essa quantidade juntamente com a string resultante.