

## Laboratório - ponteiros

### Instruções:

Todas as funções devem estar em um único arquivo aula1.c. Submeter no AVA até o final da aula.

**1** - Escreva um programa que declare um inteiro, um real e um char, e ponteiros para inteiro, real, e char. Associe as variáveis aos ponteiros (use &). Modifique os valores de cada variável usando os ponteiros. Imprima os valores das variáveis antes e após a modificação.

**2** - Escreva um programa que contenha duas variáveis inteiras. Compare seus endereços e exiba o maior endereço.

**3** - Dado o programa abaixo que lê pares de números separados por espaço e imprime os números em sua ordem original quando o primeiro é menor ou igual ao segundo e em ordem inversa quando o primeiro é maior do que o segundo. Implemente a função que faz a troca dos números de duas variáveis quando apropriado. Escolha o cabeçalho de função apropriado para a resolução desse problema. Perceba que o programa principal (função main) não deve ser alterado, exceto a passagem de parâmetros para a função.

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

//Defina e implemente TrocaInterios

int main(){
    int n1, n2;

    while ( scanf("%d %d", &n1, &n2) == 2 ){
        if ( TrocaInterios( /*PASSE OS PARAMETROS NECESSARIOS
AQUI*/ ) ){
            printf("TROCADO: %d %d\n", n1, n2);
        } else {
            printf("ORIGINAL: %d %d\n", n1, n2);
        }
    }
    return 0;
}
```

Entrada: uma lista de pares de números separadas por espaço, um par em cada linha.

Saída: "ORIGINAL:" seguido pelos números na ordem original, ou "TROCADO:" seguido pelos números com a ordem trocada.

Exemplo de Entrada:

```
1 1
2 1
1 2
-100 100
100 -100
2345 5324
234 12
0 0
1000 0
```

Exemplo de Saída:

```
ORIGINAL: 1 1
TROCADO: 1 2
ORIGINAL: 1 2
ORIGINAL: -100 100
TROCADO: -100 100
ORIGINAL: 2345 5324
TROCADO: 12 234
ORIGINAL: 0 0
TROCADO: 0 1000
```

**4** - Crie uma função que receba três parâmetros: um array, o tamanho do array e um valor do mesmo tipo do array. A função deverá preencher os elementos de array com esse valor. Não utilize índices para percorrer o array, apenas aritmética de ponteiros.

**6** - Implemente uma função que receba como parâmetro um vetor de números inteiros (vet) de tamanho n e inverta a ordem dos elementos armazenados nesse vetor. Essa função deve obedecer ao protótipo:

```
void inverte (int n, int* vet);
```

**7** - Elabore uma função que receba duas strings como parâmetros e verifique se a segunda string ocorre dentro da primeira. Use aritmética de ponteiros para acessar os caracteres das strings.