LISTA DE EXERCÍCIOS PONTEIROS

1.- Crie um programa em C que declara dois números inteiros e dois caracteres do tipo char (todos devidamente inicializados – inteiros como 1 e 2 – char como a E b).

Em seguida, mostre o VALOR de cada variável, bem como seu ENDEREÇO

Depois, altere os valores das variáveis e mostre novamente o VALOR e ENDEREÇO de cada variável desta (2112 E 666; ‘A’ e ‘B’)

2.- 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.

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

3. Escreva um programa que contenha duas variáveis inteiras. Leia essas variáveis do teclado. Em seguida, compare seus endereços e exiba o conteúdo do maior endereço.

4.- Considere a seguinte declaração: int A, \*B, \*\*C, \*\*\*D. Escreva um programa que leia a variável A e calcule e exiba o dobro, o triplo e o quadruplo desse valor utilizando apenas os ponteiros B, C e D. O ponteiro B deve ser usada para calcular o dobro, C o triplo e D o quadruplo.

5.- Crie um programa que contenha um array de float contendo 10 elementos. Imprima o endereço de cada posição desse array.

6.- Crie um programa que contenha uma matriz de float contendo 3 linhas e 3 colunas. Imprima o endereço de cada posição dessa matriz.

7.- Crie um programa que contenha um array de inteiros contendo 5 elementos. Utilizando apenas aritmética de ponteiros, leia esse array do teclado e imprima o dobro de cada ´ valor lido.

8.- Crie um programa que contenha um array contendo 5 elementos inteiros. Leia esse array do teclado e imprima o endereço das posições contendo valores pares.

9.- A variável que representa um vetor pode ser vista como um ponteiro.

O que o código a seguir vai imprimir?

1 int v[10] = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 };

2 printf("\*v = %d", \*v);

\*v = 1

\*v imprime o conteúdo (do tipo int) que está no endereço de memória v, que é exatamente igual a v[0].

v[0] é equivalente a \*v, que é equivalente a \*(v+0)

v[4] é equivalente a \*(v+4)

Aritmética de ponteiros: Ao somar 4 em um ponteiro do tipo int\*, estamos “pulando” 4 inteiros. Assim, podemos utilizar indexação (v[4]) ou aritmética de ponteiros (\*(v+4)) para ler/escrever na memória.



