

1. Indique qual será a saída dos programas abaixo (faça a resolução manual e depois implemente os programas para certificar-se do resultado).

a)

```
#include <stdio.h>
int main() {
    int a = 10, b = 20;
    int c = 30, d = 40;
    int *p1 = &c, *p2 = &d;
    a = c + *p1;
    b = b - *p2;
    c = b - *p1;
    c = *p2;
    d = *p1;
    printf("Valor de a,b,c,d:
    %d,%d,%d,%d\n",a,b,c,d);
}
```

b)

```
#include <stdio.h>
int main() {
    int a = 10, b = 20;
    int *p1 = &a, *p2 = &b, *p3;
    *p1 = b + 1;
    *p2 = a + 1;
    p3 = p1;
    printf("%d\n", *p1);
    printf("%d\n", *p2);
    printf("%d\n", *p3);
    printf("%d\n", *(&a));
    printf("%d\n", *(*(&p2)));
}
```

2. Considerando que p é um ponteiro para um inteiro, explique a diferença entre:
a) p++; b) (*p)++; c) *(p++);
3. Assumindo que v[] é um vetor do tipo int, qual/quais das seguintes expressões referenciam o valor do terceiro elemento (elemento de índice 2) de v?
a) *(v + 2); b) *(v + 4); c) v + 4; d) v + 2; e) v++;
4. Qual o valor de y no final do programa? Tente primeiro descobrir e depois verifique no computador o resultado. A seguir, escreva um /* comentário */ em cada comando de atribuição explicando o que ele faz e o valor da variável à esquerda do '=' após sua execução.

```
#include <stdio.h>
int main() {
    int y, *p, x;
    y = 0;
    p = &y;
    x = *p;
    x = 4;
    (*p)++;
    x--;
    (*p) += x;
    printf ("y = %d\n", y);
}
```

5. Escreva um programa que leia um número inteiro positivo n seguido de n números inteiros e imprima esses n números em ordem inversa. Por exemplo, ao receber [5 222 333 444 555 666] o seu programa deve imprimir 666 555 444 333 222. O seu programa deve manipular o vetor somente através dos ponteiros de cada posição.
6. Usando passagem de parâmetros por referência, escreva uma função CALCULA que:
- Receba como parâmetros duas variáveis inteiras, X e Y;
 - Retorne em X a soma de X e Y;

- Retorne em Y a subtração de X e Y.
7. Faça um programa em C que recebe como parâmetro uma medida de tempo em minutos (por exemplo, 625 minutos) e apresenta o mesmo valor no formato HH:MM (10:25 para o exemplo). Crie uma função para fazer a conversão e utilize passagem de parâmetros por referência para possibilitar que ela retorne a quantidade de horas e de minutos;
 8. Crie um programa ler as coordenadas (x1, y1, x2, y2, x3, y3) de 3 pontos em um plano cartesiano e calcule o comprimento de cada lado do triângulo (a, b, c) através de uma função. Use passagem por referência para que a função retorne os comprimentos dos 3 lados do triângulo;
 9. Altere o programa proposto na questão 8 para representar os pontos através de uma struct com nome Ponto2d e os lados do triângulo através de uma struct denominada Triangulo.
 10. Altere a função que calcula a distância entre os pontos implementada na questão 2 da lista de exercícios sobre estruturas para que ela receba como parâmetros os ponteiros para as estruturas ao invés de uma cópia das mesmas;
 11. Crie uma função chamada soma_vetor com a estrutura abaixo:
 - Parâmetros:
 - Um ponteiro p para um vetor de elementos inteiros;
 - Um inteiro t contendo o tamanho do vetor p;
 - Retorno: um valor inteiro representando a soma dos elementos do vetor;
 12. Crie uma função chamada concatena_vetores com a estrutura abaixo:
 - Parâmetros:
 - Um ponteiro p1 para um vetor de elementos inteiros;
 - Um inteiro t1 contendo o tamanho do vetor p1;
 - Um ponteiro p2 para um vetor de elementos inteiros;
 - Um inteiro t2 contendo o tamanho do vetor p2;
 - Retorno: um ponteiro para um novo vetor contendo todos os elementos de p1 e p2;