

Questão 1. Veja o código:

```
void f(int a, int b) {
    int c = a;
    a = b;
    b = c;
}

void g(int *c, int n) {
    c = malloc(sizeof(int) * n);
}

void h(int *h) {
    *h = 100;
}

int main() {
    int a = 1, b = 2, c = 3;
    int **d;
    d = malloc(sizeof(int*));
    f(a, b);
    g(d[1], 1);
    h(*d);
    free(d);
    printf("%d, %d, %d", a, c, d[1][1]);
    return 0;
}
```

- (a) É possível determinar a saída do programa (ou parte dela)? Qual?
- (b) Existe *memory leak*?
- (c) Modifique para que toda memória alocada seja liberada e a saída seja a esperada: 2, 3, 100.

Questão 2. *Matriz linearizada:* Uma outra maneira de armazenar um conjunto de pontos é salvar todas as coordenadas em uma matriz. Assim, cada ponto é representado por uma linha ou por uma coluna.

Escreva um programa que calcule o centroide de um conjunto de pontos e calcule a soma das distâncias de cada ponto para o centroide. Serão lidos em ordem a dimensão dos pontos, o número de pontos e as coordenadas em ordem de cada ponto.

Você deve utilizar alocação dinâmica de memória e uma matriz para armazenar os pontos. Assim, a matriz deverá ser representada na memória como um vetor. Como isso pode ser feito?

Questão 3. Reflita e responda:

- (a) Qual a diferença entre passagem por valor e referência?
- (b) Quando é vantajoso passar registros (struct) por referências?
- (c) E quando é melhor usar passagem por valores?