

Aula Prática 9 – 24/05/2018

Preparem os exercícios de forma que:

1. Os arquivos utilizem a extensão **“.c”**
2. Não seja utilizada função **system(“pause”)**
3. A função **printf** deve ser utilizada apenas para imprimir a saída do programa.

Atenção: Para esta lista de exercícios é necessário utilizar funções conforme pedem as atividades, caso o contrário, a nota será **0**.

1) Faça um programa que leia duas matrizes, A e B, de qualquer dimensão e calcule e imprima o produto destas matrizes. O programa deve funcionar da seguinte forma:

1. Solicitar a leitura de 2 valores inteiros, relacionados com o número de linhas e colunas respectivamente da matriz A;
2. Solicitar a leitura de 2 valores inteiros, relacionados com o número de linhas e colunas respectivamente da matriz B;
3. Verificar se a multiplicação é possível, se não for deve imprimir a mensagem “Operacao invalida!\n”
4. Se a multiplicação for possível, o programa deve solicitar a leitura da matriz A, depois da matriz B e então calcular e imprimir o produto AB. A impressão da matriz deve usar a função mostrada abaixo:

```
void imprimir_matriz(int **m, int l, int c){ // l = linha, c = coluna
    int i, j;
    for (i=0; i<l; i++){
        for (j=0; j<c; j++){
            printf("%d\t", m[i][j]);
        }
        printf("\n");
    }
}
```

As matrizes devem armazenar valores inteiros, e a leitura deve preencher linha por linha.

- **Entrada:**
 - quatro números inteiros, em seguida os valores das matrizes se for possível a multiplicação.
- **Saída:** uma das seguintes mensagens:
 - “Operacao invalida!\n”
 - os valores da matriz usando a função imprimir_matriz.

2) Faça um programa que leia dois vetores de tamanhos distintos, e depois retorne a soma desses vetores. Seu programa deve funcionar da seguinte forma:

1. Leia um valor **N** inteiro, que vai representar o tamanho do primeiro vetor, logo depois disso, leia outro valor inteiro **M** que vai representar o tamanho do segundo vetor;
2. Após isso, leia **N** valores inteiros, armazenando-os no primeiro vetor, e em seguida leia **M** valores inteiros, armazenando-os no segundo vetor;
3. Como **M** e **N** podem ter tamanhos diferentes, a soma só será possível onde houver elementos nos dois vetores. Onde não houver elementos em um dos vetores, o valor a ser colocado no vetor resultante, deve ser o mesmo valor disponível no vetor que possuir valores.
4. Após isso, imprimir todos os valores do vetor, separados por uma nova linha:
"%d\n"

Exemplo

• Entrada:	<ul style="list-style-type: none"> • 2 4 • 9 5 • 5 5 10 2
• Saída:	<ul style="list-style-type: none"> • 14 • 10 • 10 • 2

3) Faça um programa que contenha duas estruturas, uma estrutura do tipo **Ponto** e outra do tipo **Retangulo**. Assuma sempre que os lados do retângulo são paralelos ao eixo-x e ao eixo-y. Você deve implementar duas funções, uma que calcule a área do retângulo, e outra que teste se um dado ponto está em um retângulo. Seu programa deve funcionar da seguinte forma:

A estrutura **Ponto** deve representar com números de ponto flutuante as coordenadas **x** e **y**.

A estrutura **Retangulo** deve ser representada com duas variáveis do tipo **Ponto**, representando o canto superior direito, e o canto inferior esquerdo.

1. Seu programa deve ler um inteiro **N**, representando a quantidade de Retângulos a serem lidos. Logo após, ler **4N** números float com as informações dos pontos do retângulo, sempre ler primeiro o ponto superior direito, pra depois ler o inferior esquerdo, em um ponto, sempre ler o valor **x** primeiro.
2. Após construído todos os retângulos, imprimir a área de cada um deles com a seguinte mensagem:
"Area do retangulo %d: %.2f\n"
Onde o primeiro **%d** representa o indice do retângulo.
3. Seu programa deve então ler um inteiro **M**, representando a quantidade de pontos a serem testados. Após isso, ler **2M** números float com as

informações de **M** pontos. Para cada ponto, sempre leia o valor de **x** primeiro.

4. Para cada um dos pontos, testar se ele está dentro (borda inclusive) do último retângulo lido. E imprimir o resultado com uma das seguintes mensagens:

"O ponto %d esta dentro do retangulo %d\n"

"O ponto %d nao esta dentro do retangulo %d\n"

Onde, o primeiro **%d** representa o indice do ponto, e o segundo **%d** representa o indice do ultimo retângulo.

Exemplo:

Entrada:	2 1.3 3.1 0.2 2.8 10.9 13.2 2.5 3.2 4 2.2 2.9 7.3 5.8 11.0 17.2 18.2 11.2
Saída:	Area do retangulo 0: 0.33 Area do retangulo 1: 84.00 0 ponto 0 nao esta dentro do retangulo 1 0 ponto 1 esta dentro do rentagulo 1 0 ponto 2 nao esta dentro do retangulo 1 0 ponto 3 nao esta dentro do retangulo 1