

(TAD_gen_06) Uma empresa de computação científica contratou você para implementar, em C, uma alternativa para uma biblioteca conhecida da área. Sua primeira tarefa nesta empresa é implementar um TAD referente a matrizes. Porém, este TAD **deve suportar matrizes de qualquer tipo de dados**, sejam eles caracteres, números inteiros, números reais (precisão simples ou dupla), ou algum outro tipo que pode ser até definido pelo usuário. Isto é necessário pois setores distintos da empresa possuem necessidades diferentes.

Além de armazenar uma matriz de um tipo qualquer, seu TAD deve possuir ao menos quatro outras funções. A primeira deve imprimir uma matriz; A segunda deve tornar possível converter matriz de um tipo para outro (usando *callbacks* externos que convertem os dados); A terceira deve implementar a transposição de uma matriz (retornar a transposta da matriz, mantendo o mesmo tipo); E a quarta deve fazer a multiplicação de matrizes de um mesmo tipo (usando *callbacks* para somar e multiplicar elementos do mesmo tipo).

Desta forma, implemente um TAD genérico (e opaco) que tenha essas funcionalidades. Implemente também um TAD que representa um número complexo. Após, implemente um programa que teste as quatro funções do TAD, seguindo o padrão de entrada definido abaixo.

Um número complexo é um número que possui parte real (R) e parte imaginária (I). Ele é representado matematicamente da seguinte forma:

$$Z_1 = R_1 + i I_1,$$

onde $i = \sqrt{-1}$. Neste roteiro, vocês devem implementar apenas funções básicas de soma e multiplicação de números complexos, conforme fórmulas a seguir:

Soma ($Z_3 = Z_2 + Z_1$):

$$Z_3 = Z_2 + Z_1 = (R_1 + R_2) + i * (I_1 + I_2)$$

Multiplicação ($Z_3 = Z_2 * Z_1$)

$$Z_3 = Z_2 * Z_1 = (R_1 + i I_1) * (R_2 + i I_2) = (R_1 R_2 - I_1 I_2) + i (R_1 I_2 + R_2 I_1)$$

Para outras informações sobre números complexos, você pode consultar [este link](#)

Definição dos formatos de entrada e saída:

Entrada: Seu programa deve receber inicialmente três números numa mesma linha: número de linhas da matriz; números de colunas da matriz e o tipo da matriz (ex: 0 – inteiro, 1 – float, 2 – double, 3 – char, 4 – número complexo). Em seguida, deve receber os números da matriz, linha a linha. Se ela for do tipo char, considere que ela

deve receber apenas dígitos de 0 a 9. Se ela for do tipo complexo, cada elemento da matriz deve ser representado por uma dupla (Re,Im).

Após terminar de preencher a matriz, o usuário deve entrar com a operação desejada: 1 – Imprimir a matriz lida; 2 – Converter a matriz para o tipo complexo e posteriormente a imprimir; 3 – Converter a matriz para o tipo complexo, calcular e imprimir a multiplicação da matriz pela sua transposta: MM^T . Após, o programa deve perguntar ao usuário se deseja sair ou fazer outra operação com uma nova matriz.

Saída: Após receber as entradas, o programa deve seguir as opções do usuário até que ele decida sair do programa.

Ver exemplos de formato de entrada e saída nos arquivos fornecidos com a questão.