/\*\*  
\* UFRJ - Universidade Federal do Rio de Janeiro  
\* IM - Instituto de Matemática  
\* DMA - Departamento de Matemática Aplicada  
\*  
\* TMAB - Programação C/C++ & Banco de Dados  
\* ParteI - Programação Estruturada em C  
\*   
\* Descrição do Estudo Dirigido 03  
\* Prazo de Entrega: 17 de outubro de 2017  
\*  
\* Prof. Milton Ramirez (milton@labma.ufrj.br)  
\* Rio de Janeiro, 09 de outubro de 2017  
\*\*/

"Calculadora Matricial"

Programar uma calculadora cujos dados são matrizes ‘m’ linhas por ‘n’ colunas de dados ponto flutuante.  
A calculadora manipulará uma LISTA de matrizes sobre as quais permitirá a realização de operações matriciais, conforme descrito a seguir. O resultado deverá ser colocado como uma nova entrada na LISTA de matrizes da calculadora.  
As operações que a calculadora deve oferecer aos seus usuários são:  
1. Soma matricial: C = A + B.  
2. Multiplicação por um escalar: C = alfa\*A.  
3. Multiplicação Matricial: C = A x B.  
4. Transposição Matricial: C = Transposta(A).

As operações deverão testar a compatibilidades dos operando antes de realizar os cálculos.

A calculadora deverá também gravar em arquivo e recuperar de um arquivo uma única matriz ou a sua lista de matrizes, podendo ter comandos para zerar a lista, ou carregar uma nova lista de matrizes.

O formato do arquivo de uma matriz, ou da lista de matrizes é de escolha do programador. Sugerimos colocar o número de linhas e colunas antes da lista de elementos da matriz. Para o caso do arquivo contendo a LISTA de matrizes, sugerimos replicar o esquema anterior para cada matriz da lista, que se sucederão gravadas no arquivo, e gravar no início do arquivo o número de matrizes que o arquivo contém.

A lista de matrizes deverá ser implementada usando um ‘container’ da STL C++. Sugestão: usar “vector<CMatriz>” ou “vector<CMatriz \*>”, onde “CMatriz” é o nome da classe criada pelo programador.

As matrizes deverão ser implementadas como classes de objetos de tamanho variável, ou seja, o numero de linhas e colunas da matriz deverá ser determinado no momento da criação da matriz e o espaço de memória alocado para os seus elementos deverá ter o tamanho justo para conter apenas os seus elementos.   
Sugerimos um “vector< vector<float>>”.

O programa deverá estar separado em pelo menos dois arquivos (ou módulos). O primeiro deverá conter as rotinas de controle e visualização de dados do programa. O segundo (CMatriz.cpp) deverá conter a implementação de todas as funções membro da classe “CMatriz”. Programe também um arquivo ‘head’ <CMatriz.h> para essa segunda parte, contendo a definição da classe “CMatriz”. Pode-se usar o wizard do CodeBlocks para essa tarefa.

Por fim, a rotina de “Soma Matricial” deverá ser implementada como uma sobrecarga do operador de soma do C++, ou seja:  
CMatriz CMAtriz::operator +(CMatriz B) {...}

A sobrecarga de operadores será assunto da aula de 10 de outubro de 2017.