ISEL	ESTRUTURAS DE DADOS E ALGORITMOS
Sumário: Ficha 2	Duração: 2 Aulas

PARTE I

Esta ficha consiste na criação de uma classe para representar uma **matriz de reais** com as operações típicas que se realizam com matrizes. Os elementos da matriz devem ser guardados num *array* dinâmico de duas dimensões de acordo com a ordem da matriz.

Ficheiro Matriz.h

```
#pragma once
// Definição da classe Matriz que contem as operações
// tipicas de matrizes

class Matrix {
private:
	float **elems;
	int nLines;
	int nCols;

	// Funções privadas
	void Delete();                     // Apagar a memória dinâmica criada para a matriz
	void Create(int lines, int cols); // Criar a memória dinâmica e inicializar a matriz

public:
	Matrix();                   // Construtor de defeito
	Matrix(const Matrix& m1);   // Construtor Copy
	Matrix(int lines, int cols);
	~Matrix();           // Destrutor

	bool   CanMultiply(const Matrix* pm);
	bool   CanAdd(const Matrix* pm);

	const Matrix& operator = (const Matriz& m1);
	Matrix operator + (const Matriz& m1);
	Matrix operator * (const Matriz& m1);
	const Matrix& operator += (int k);     // Soma cada elemento da matriz com o valor k
	const Matrix& operator *= (int k);     // Multiplica cada elemento da matriz por k

	void Transpose();
	bool ReadFromFile(char* file_name);
	bool SaveToFile(char* file_name);
	void Output();

	Matrix GetLU();
};
```

A classe deve designar-se por ***Matrix*** e possuir os seguintes atributos privados:

- ***elems*** – elementos da matriz.
- ***nLines*** – número de linhas da matriz.
- ***nCols*** – número de colunas da matriz.

A classe deve conter:

- Um construtor que recebe a ordem da matriz e cria uma matriz com elementos a zero.
- Um construtor cópia que cria uma nova matriz a partir de outra já existente.
- Um construtor por omissão (a ordem da matriz é definida posteriormente).
- Um destrutor para eliminar os dados dinâmicos da classe.

E os seguintes métodos:

- ***ReadFromFile*** – Lê os dados da matriz de um ficheiro(1).
- ***SaveToFile*** – Grava os dados da matriz para um ficheiro[0].
- ***Output*** – Escreve no ecrã o conteúdo da matriz.
- ***CanAdd*** – Verifica se duas matrizes podem ser somadas.
- ***CanMultiply*** – Verifica se duas matrizes podem ser multiplicadas.
- ***operador atribuição* (=)** – atribuição de matrizes(3).
- ***operador soma* (+)** – soma de matrizes(4) e soma dos elementos da matriz por um valor constante.
- ***operador produto* (*)** – produto de matrizes(4) e produto dos elementos da matriz por uma constante.
- ***Transpose*** – Deve realizar a operação de transpor a própria matriz, i.e., deve trocar as linhas pelas colunas.
- ***GetLU*** – Função explicada na parte II deste trabalho.

Deve realizar uma função main que chame estas funções definidas na classe Matriz.

Notas:

(1) Esta função deve receber como parâmetro de entrada o nome do ficheiro e devolver um valor booleano para indicar se a leitura foi correta ou não. O formato do ficheiro deve ser o seguinte:

Ficheiro Matriz.txt

```
# Comentário       // linhas iniciadas com # são consideradas comentário
2  2          // 1ª linha com o nº de linhas e colunas da matriz
1             // 2ª linha e seguintes com cada elemento da matriz
2
-3
0
```

(2) Esta função deve gravar um ficheiro de texto com o formato indicado em (1).

(3) O conteúdo da matriz da direita é copiado para a matriz da esquerda. Se a matriz da esquerda contiver dados, estes devem ser removidos previamente.

(4) As matrizes devem ser de dimensões compatíveis com a operação a realizar.

PARTE II

Qualquer matriz quadrada A pode ser decomposta no produto $A = LU$ onde L e U são as matrizes triangulares inferior e superior respetivamente.

Para matrizes 3x3 tem-se o seguinte exemplo:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \qquad L = \begin{bmatrix} 1 & 0 & 0 \\ l_{21} & 1 & 0 \\ l_{31} & l_{32} & 1 \end{bmatrix} \qquad U = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

Fazendo a operação $A = LU$

$$A = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ u_{11}l_{21} & u_{12}l_{21} + u_{22} & u_{13}l_{21} + u_{23} \\ u_{11}l_{31} & u_{12}l_{31} + u_{22}l_{32} & u_{13}l_{31} + u_{23}l_{32} + u_{33} \end{bmatrix}$$

Eliminando os elementos debaixo do pivot $(LU)_{11}$

$(Row_2 - L_{21}Row_1) \rightarrow Row_2$ e assim elimina-se $(LU)_{21}$

$(Row_3 - L_{31}Row_1) \rightarrow$ Row3 e assim elimina-se $(LU)_{31}$

$$A' = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & u_{22}l_{32} & u_{23}l_{32} + u_{33} \end{bmatrix}$$

Aplicando novamente esta técnica de eliminação, conhecida como eliminação de Gauss, elimina-se os elementos debaixo do pivot $(LU)_{22}$

$(Row_3 - L_{22}Row_2) \rightarrow Row_3$ e assim elimina-se $(LU)_{32}$

$$A'' = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{bmatrix}$$

A matriz U é aquela que resulta da eliminação de Gauss. É prática corrente guardar numa só matriz tanto a matriz triangular superior como a matriz triangular inferior. É de notar que os elementos da diagonal principal da matriz L não são necessários guardar uma vez que são 1.

$$[L\backslash U] = \begin{bmatrix} u_{11} & u_{12} & u_{13} \\ l_{21} & u_{22} & u_{23} \\ l_{31} & l_{32} & u_{33} \end{bmatrix}$$

```
Pseudo-Código: Factorização A = L.U

// matriz A(nxn)
// Eliminação de Gauss
Ciclo nas colunas (k) : k = 0, n-2
    Ciclo nas linhas debaixo da linha pivot:
          i = k+1, n-1
     - Para cada linha:
            Calcular o factor multiplicador:
                    A(i,k)/A(k,k)
     - Transformar linha i:
               Somente elementos (i, k+1:n) são guardados
     - Guardar factor mutiplicadores em A(i,k)
```

Pretende-se que altere a classe Matriz da ficha anterior para incluir um método que efectue a decomposição LU e devolva uma matriz como resultado com essa decomposição.

Exemplo:

Quer-se encontrar a decomposição LU da seguinte matriz:

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 2 & -4 & 6 \\ 3 & -9 & -3 \end{bmatrix}$$

1ª coluna k = 0

2ª linha i = 1

$$l_{10} = \frac{a_{10}}{a_{00}} = 2$$

$$row_1 - l_{10} row_1 = [0 \quad -8 \quad 0]$$

3ª linha i = 2

$$l_{20} = \frac{a_{20}}{a_{00}} = 3$$

$$row_2 - l_{20} row_1 = [0 \quad -15 \quad -12]$$

$$A' = \begin{bmatrix} 1 & 2 & 3 \\ \mathbf{2} & -8 & 0 \\ \mathbf{3} & -15 & -12 \end{bmatrix}$$

2ª coluna k = 1

3ª linha i = 2

$$l_{21} = \frac{a_{21}}{a_{11}} = \frac{15}{8}$$

$$row_3 - l_{21} row_2 = [0 \quad 0 \quad -12]$$

$$[L \backslash U] = \begin{bmatrix} 1 & 2 & 3 \\ \mathbf{2} & -8 & 0 \\ \mathbf{3} & \mathbf{15/8} & -12 \end{bmatrix} \quad A = \begin{bmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 15/8 & 1 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 \\ 0 & -8 & 0 \\ 0 & 0 & -12 \end{bmatrix}$$