Equipo 1

Método para multiplicar objetos de la clase Matriz del proyecto que les compartí en github.

En la clase Matriz (en el archivo Matriz.h), agregar (en la sección publica) el método

Matriz operator\*(Matriz other);

Es decir, quedará así:

/\*\*Matriz.h\*/

class Matriz {

// …

Matriz operator\*(Matriz other);

// …

};/\*end class Matriz\*/

Y deberán resolver el método operator\*(), en el archivo (incluyendo, pero no limitado a este archivo) Matriz.cpp

/\*\*Matriz.cpp\*/

//…

Matriz Matriz::operator\*(Matriz other){

// ESCRIBA SU CODIGO AQUI

}/\*end Matriz::operator\*()\*/

//…

Cabe mencionar, que también deberán agregar en los archivos Entrada.h, Escalar.h, Escalar.cpp, Polinomio.h y Polinomio.cpp, los prototipos de métodos y resolución de los métodos que sean necesarios para poder programar la sobrecarga del operador \* para los objetos de la clase Matriz.

Equipo 2

Método para calcular el determinante de un objeto de la clase Matriz que les compartí en mi repositorio de github.

Es decir, en la clase Matriz (archivo Matriz.h) deberán agregar (en la sección pública) el método

/\*\*Matriz.h\*/

Class Matriz {

// …

Escalar\* det();

// …

};/\*end class Matriz\*/

Y deberán resolver el método det() en el archivo de implementación (incluyendo, pero no limitado a este archivo) Matriz.cpp. Es decir, en el archiv Matriz.cpp, deberán codificar

/\*\*Matriz.cpp\*/

Escalar\* Matriz::det(){

//ESCRIBA SU CODIGO AQUI

}/\*end Matriz::det()\*/

// …

Cabe mencionar, que también deberán agregar en los archivos Entrada.h, Escalar.h, Escalar.cpp, Polinomio.h y Polinomio.cpp, los prototipos de métodos y resolución de los métodos que sean necesarios para poder programar la función det() para los objetos de la clase Matriz.

Equipo 3

Método para sustituir la variable de los polinomios que son elementos de los objetos de clase Matriz. En la clase Matriz (en el archivo Matriz.h), agregar (en la sección publica) el método

Matriz substituir(int x);

Es decir, el archivo Matriz.h deberá quedar como sigue:

/\*\*Matriz.h\*/

class Matriz {

// …

Escalar\* substituir(int x);

// …

};/\*end class Matriz\*/

Y deberán resolver el método Matriz::substituir(int x) en el archivo (incluyendo, pero no limitado a este archivo ) Matriz.cpp

Para que en el archivo Matriz.cpp que algo como lo siguiente:

/\*\*Matriz.cpp\*/

// …

Matriz Matriz::substituir(int x){

// ESCRIBA SU CODIGO AQUÍ

}/\*end Matriz::substituir()\*/

Cabe mencionar, que también deberán agregar en los archivos Entrada.h, Escalar.h, Escalar.cpp, Polinomio.h y Polinomio.cpp, los prototipos de métodos y resolución de los métodos que sean necesarios para poder programar la función substituir(int x) para los objetos de la clase Matriz.

Lo siguiente, es solo un ejemplo:

%FontSize=11
%TeXFontSize=11
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
A(s)=\left[\begin{array}{cc}
s+1&s^{2}\\
2&s+2
\end{array}\right]
\]
\end{document}

Al sustituir s=2, se obtiene

%FontSize=11
%TeXFontSize=11
\documentclass{article}
\pagestyle{empty}
\begin{document}
\[
\left[\begin{array}{cc}
3&4\\
2&4
\end{array}\right]
\]
\end{document}

Por ejemplo el código en la clase cliente podría ser:

Matriz A(ROWS,COLS,arr);

Matriz B=A.substituir(2);

std::cout<<B<<std::endl;

EQUIPO 4

El sistema trabaja con matrices cuyas entradas pueden ser números enteros (Escalar\*), o bien, pueden ser polinomios de una variable (Polinomio\*). Es decir, Algunos elementos de la matriz son Escalar\* y otros son Polinomio\*. Sin embargo, desde el punto de vista matemático, los elementos de la matriz que son enteros (Escalar\*) podemos pensarlos como polinomios de grado 0. Así que el método que les asignaré al equipo 4 es el siguiente: Escriban un método llamado to\_PolinomioPtr(), que lo que deberá hacer es transformar la Matriz en un objeto de la clase Matriz en la cual, todas las entradas sean de tipo Polinomio\*.