**Primer Examen Parcial 2023**

***Parte prácica***

Escriba un programa en lenguaje **C++** que permita ejecutar el conjunto de sentencias o tareas mostradas en el siguiente código:

**int** main()

{

CMatriz original;

original.cargar("colores.bin");

**cout** << "Matriz original:" << **endl** << original << **endl**;

CMatriz copia(original),

suma;

suma = original;

suma+= (copia + 5);

**cout** << "Matriz suma:" << **endl** << suma <<**endl**;

CPantalla pantalla(original);

**cout** << "Pantalla original:" << **endl** << pantalla << **endl**;

pantalla.ajustarColor(3);

**cout** << "Pantalla con ajuste de color:" << **endl** << pantalla << **endl**;

pantalla.borrarVerde();

**cout** << "Pantalla sin componente verde:" << **endl** << pantalla << **endl**;

pantalla.reforzarRojo(0.35);

**cout** << "Pantalla modificada (rojo reforzado):" << **endl** << pantalla << **endl**;

//generar la misma informacion en un archivo de salida: pantalla.txt

CColor color(pantalla.getPtr()[pantalla.getFilas()\*pantalla.getColumnas()-1]);

**cout** << "El color del ultimo punto de la pantalla es: " << **endl**

<< color << **endl**;

**return** 0;

}

El programa debe cumplir con las siguientes consignas:

1) El método cargar de la clase CMatriz debe recibir como parámetro el nombre del archivo binario donde se encuentran almacenados los valores de la misma. El archivo contiene todos valores de tipo unsigned int con la siguiente secuencia:

| cantidad\_filas | cantidad\_columnas | todos\_los\_valores |

El método debe estar optimizado para realizar la lectura con la menor cantidad de sentencias posible.

Los valores deben imprimirse en hexadecimal, incluyendo el prefijo 0x y con un ancho de columna adecuado para facilitar la visualización de los mismos.

2) El objeto pantalla es una matriz cuyas componentes representan el color de cada pixel en formato:

0xAARRGGBB 🡪 AA es la componente *alpha*

RR es la componente roja (*red*)

GG es la componente verde (*green*)

BB es la componente azul (*blue*)

La clase pantalla incluye además funciones miembro que son capaces de modificar cada una de las componentes de color por separado. En particular:

1. ajustarColor(numero) suma el numero indicado a la componente *alpha* de todos los colores.
2. borrarVerde() establece la componente *green* a 0 en todos los colores.
3. reforzarRojo(porcentaje) aumenta la componente *red* de todos los colores en el porcentaje indicado. Debe tenerse en cuenta que cada componente puede adoptar un valor máximo de 0xFF, por lo que si el aumento excede de este valor se debe ajustar la componente al ese máximo.

3) Todos los datos deben ser capaces de persistirse en un archivo de texto.

4) La clase CColor debe poder imprimir un color con el siguiente formato:

El color del ultimo punto de la pantalla es:

componente azul: valor\_azul

componente verde: valor\_verde

componente rojo: valor\_rojo

componentealpha: valor\_alpha

5) Se debe realizar una correcta modularización del código, y el programa debe construirse (build) sin errores (errors) ni advertencias (warnings).

6) Recuerde gestionar adecuadamente el uso de la memoria dinámica utilizando las funciones proporcionadas por el lenguaje a tal efecto.

Hint: para considerar los métodos que deben implementar las distintas clases considere las sentencias usadas en el main.