**Hecho por: Mateo Pineda Perez  
  
Introducción**

Este reto consiste en recuperar una imagen original que fue modificada a través de varias transformaciones que alteran sus bits. El problema es que no sabemos qué transformaciones se hicieron ni en qué orden. Para resolverlo, debemos usar lo que hemos aprendido en programación en C++, especialmente sobre cómo manejar bits, memoria y arreglos, sin usar herramientas avanzadas como la STL.

Desarrollo

La imagen original (IO) pasó por una serie de cambios que alteraron sus bits. Algunas de estas transformaciones incluyen:

1. Operación XOR con una imagen aleatoria (IM), que cambia los bits de la imagen original.
2. Rotación de bits en los canales de color (RGB), que mueve los bits a la derecha o izquierda.
3. Enmascaramiento con una máscara de color (M), que suma los valores de color de la imagen con los de la máscara.

1.1. Entender los componentes del problema

* Imagen final transformada (Io): Es el resultado de aplicar múltiples transformaciones a la imagen original.
* Imagen aleatoria (IM): Se usa en operaciones XOR.
* Máscara (M): Se usa en enmascaramientos.
* Archivos .txt (M1.txt, M2.txt, ...): Contienen información sobre los enmascaramientos aplicados.

1.2. Transformaciones aplicadas

* Operación XOR (XOR(Io, IM)): Esta operación cambia los bits de la imagen y puede revertirse aplicando XOR nuevamente con la misma imagen aleatoria (IM).
* Rotación de bits (Rotación 3 bits a la derecha): Esta operación se revierte haciendo una rotación a la izquierda.
* Enmascaramiento (Suma con M): El enmascaramiento suma los valores de color de la imagen con los de la máscara, pero puede revertirse restando la máscara.

1.3. Objetivo del algoritmo

El objetivo es determinar el orden de las transformaciones usando los archivos .txt. A partir de ahí, debemos aplicar las operaciones inversas en orden opuesto para recuperar la imagen original.

Al final, tenemos la imagen modificada (ID), que viene acompañada de:

* La imagen IM.
* La máscara M.
* Varios archivos .txt que muestran cómo cambió la imagen después de cada transformación (excepto la última).

Cada archivo .txt tiene la posición de inicio y los valores RGB que resultaron de sumar una parte de la imagen y la máscara. Estos datos nos sirven para saber si una transformación fue aplicada correctamente. Debemos probar diferentes combinaciones de transformaciones, simular los resultados y compararlos con los archivos .txt hasta encontrar la secuencia correcta.

Lo complicado aquí es que no sabemos qué transformaciones se usaron ni en qué orden. Así que, necesitamos probar diferentes opciones, hacer simulaciones y comparar los resultados con los archivos para descubrir la secuencia correcta.

Conclusión

Este reto nos ayuda a poner en práctica muchas cosas que hemos aprendido en el curso, como trabajar con arreglos, memoria dinámica y operaciones con bits. También nos da la oportunidad de mejorar nuestras habilidades para analizar, probar y corregir soluciones, habilidades que son muy útiles para resolver problemas en programación e ingeniería.

Para tener éxito, necesitamos una solución organizada que simule cada transformación, verifique si los resultados coinciden con los archivos y siga los pasos en el orden adecuado. Este desafío no solo mide nuestra habilidad técnica, sino también nuestra capacidad para resolver problemas nuevos y complejos con los recursos que ya tenemos.