```
/*
Name: Albara Mehene
Date: 10/1/2016
Computing IV

*/

#include <SFML/System.hpp>
#include <SFML/Window.hpp>
#include <SFML/Graphics.hpp>

#include "LFSR.hpp"

sf::Image transform(sf::Image picture, LFSR lfsr);

int main(int argc, char* argv[])
{
        if(argc < 5){

                        std::cout << "input-file.png, output-file.png, seed, tap" << std::endl;
                        return -1;
        }

        std::string input = argv[1];
        std::string output = argv[2];
        std::string i_seed = argv[3];
        int i_tap = atoi(argv[4]);

        sf::Image image;
        if (!image.loadFromFile(input))
                return -1;

        sf::Image image_e = image;
        sf::Image temp_e;

        //Pass the seed and tap function
        LFSR l(i_seed, i_tap);
        temp_e = transform(image_e, l);


        sf::Vector2u size = image.getSize();
        sf::Vector2u size2 = temp_e.getSize();
        sf::RenderWindow window(sf::VideoMode(size.x, size.y), "Picture1");
        sf::RenderWindow window1(sf::VideoMode(size2.x, size2.y), "Picture2");




        sf::Texture texture;
        texture.loadFromImage(image);

        sf::Texture texture_e;
        texture_e.loadFromImage(temp_e);

        sf::Sprite sprite;
        sprite.setTexture(texture);

        sf::Sprite sprite_e;
        sprite_e.setTexture(texture_e);

        while (window.isOpen() && window1.isOpen())
        {
                sf::Event event;
                while (window.pollEvent(event))
                {
                        if (event.type == sf::Event::Closed)
                                window.close();
                }
                while (window1.pollEvent(event))
                {
                        if (event.type == sf::Event::Closed)
                                window1.close();
                }


                window.clear(sf::Color::White);
                window1.clear(sf::Color::White);
                window.draw(sprite);
                window1.draw(sprite_e);
                window.display();
                window1.display();
        }

        // fredm: saving a PNG segfaults for me, though it does properly
        //   write the file
        if (!temp_e.saveToFile(output))
                return -1;

        return 0;
}

sf::Image transform(sf::Image picture, LFSR lfsr){
        // p is a pixel
        sf::Color p;
        int temp;
        sf::Vector2u size = picture.getSize();

        // create photographic negative image of upper-left 200 px square
        for (unsigned int x= 0; x < size.x; x++) {
                for (unsigned int y = 0; y < size.y; y++) {
                        p = picture.getPixel(x, y);

                        temp = lfsr.generate(8);
                        p.r = p.r ^ temp;

                        temp = lfsr.generate(8);
                        p.g = p.g ^ temp;

                        temp = lfsr.generate(8);
                        p.b = p.b ^ temp;

                        picture.setPixel(x, y, p);
                }
        }
        return picture;
}




```