```
//
//  main.cpp
//  ps2b
//
//  Created by Jingxian Shi on 2/9/18.
//  Copyright Â© 2018 Jingxian Shi. All rights reserved.
//

#include <iostream>
#include <SFML/System.hpp>
#include <SFML/Window.hpp>
#include <SFML/Graphics.hpp>
#include <string>
#include "LFSR.hpp"

sf::Image transform(sf::Image image, LFSR l);

int main(int argc, const char* argv[])
{
//    if(argc != 5)
//    {
//        std::cout << "Wrong number of arguments" << std::endl;
//        return -1;
//    }

    std::string input_image = argv[1];
    std::string output_image = argv[2];
    std::string seed = argv[3];
    int tap = atoi(argv[4]);

    sf::Image image, encrypted_image;
    if (!image.loadFromFile(input_image))
    {
        return -1;
    }
    if (!encrypted_image.loadFromFile(input_image))
    {
        return -1;
    }

    sf::Color p;
    sf::Vector2u size = image.getSize();
    LFSR l(seed, tap);

    for (int x = 0; x< size.x; x++) {
        for (int y = 0; y< size.y; y++) {
            p = encrypted_image.getPixel(x, y);
            p.r ^= l.generate(8);
            p.g ^= l.generate(8);
            p.b ^= l.generate(8);
            encrypted_image.setPixel(x, y, p);
        }
    }

    sf::RenderWindow window1(sf::VideoMode(size.x, size.y), "Original");
    sf::RenderWindow window2(sf::VideoMode(size.x, size.y), "PhotoMagic");

    sf::Texture texture1, texture2;
    texture1.loadFromImage(image);
    texture2.loadFromImage(encrypted_image);

    sf::Sprite sprite1, sprite2;
    sprite1.setTexture(texture1);
    sprite2.setTexture(texture2);

```

```
66:     while (window1.isOpen() && window2.isOpen()) {
67:         sf::Event event;
68:         while (window1.pollEvent(event)) {
69:             if (event.type == sf::Event::Closed)
70:             window1.close();
71:         }
72:         while (window2.pollEvent(event)) {
73:             if (event.type == sf::Event::Closed)
74:             window2.close();
75:         }
76:         window1.clear();
77:         window1.draw(sprite1);
78:         window1.display();
79:         window2.clear();
80:         window2.draw(sprite2);
81:         window2.display();
82:     }
83:
84:     if (!encrypted_image.saveToFile(output_image))
85:     {
86:         return -1;
87:     }
88:
89:     return 0;
90: }
91:
```