

```
1: #include <SFML/System.hpp>
2: #include <SFML/Window.hpp>
3: #include <SFML/Graphics.hpp>
4: #include <string>
5: #include <cstdlib>
6: #include "LFSR.hpp"
7:
8: void transform_method(sf::Image& image, sf::Color p, sf::Vector2u size, LFSR
1);
9:
10: int main(int argc, char* argv[])
11: {
12:     //Read in Command Line Arguments
13:     std::string user_image1 = argv[1];
14:     std::string user_image2 = argv[2];
15:     int tap = atoi(argv[4]);
16:     LFSR l(argv[3], tap);
17:
18:     //Load Images
19:     sf::Image image;
20:     sf::Image image1;
21:     if (!image.loadFromFile(user_image1))
22:     {
23:         return -1;
24:     }
25:     if (!image1.loadFromFile(user_image1))
26:     {
27:         return -1;
28:     }
29:
30:     //Create pixel variable
31:     sf::Color p;
32:
33:     //Set Size
34:     sf::Vector2u size = image1.getSize();
35:
36:     //Transform the Images
37:     transform_method(image1, p, size, l);
38:
39:     //Save Image
40:     if (!image1.saveToFile(user_image2))
41:     {
42:         return -1;
43:     }
44:
45:     //After transform
46:     sf::Image image2;
47:     if (!image2.loadFromFile(user_image2))
48:     {
49:         return -1;
50:     }
51:
52:     //Generate Windows
53:     sf::RenderWindow window1(sf::VideoMode(size.x, size.y), "Input");
54:     sf::RenderWindow window2(sf::VideoMode(size.x, size.y), "Output");
55:
56:     //Load Textures
57:     sf::Texture texture1;
58:     texture1.loadFromImage(image);
59:     sf::Texture texture2;
60:     texture2.loadFromImage(image2);
```

```
61:
62:     //Set Sprites
63:     sf::Sprite spritel;
64:     spritel.setTexture(texture1);
65:
66:     sf::Sprite sprite2;
67:     sprite2.setTexture(texture2);
68:
69:
70:
71:     //Generate Windows
72:     while (window1.isOpen() && window2.isOpen())
73:     {
74:         sf::Event event;
75:         while (window1.pollEvent(event))
76:         {
77:             if (event.type == sf::Event::Closed)
78:             {
79:                 window1.close();
80:             }
81:         }
82:
83:         while (window2.pollEvent(event))
84:         {
85:             if (event.type == sf::Event::Closed)
86:             {
87:                 window2.close();
88:             }
89:         }
90:
91:         window1.clear(sf::Color::White);
92:         window1.draw(spritel);
93:         window1.display();
94:
95:         window2.clear(sf::Color::White);
96:         window2.draw(sprite2);
97:         window2.display();
98:     }
99:
100:     return 0;
101: }
102:
103: void transform_method(sf::Image& image, sf::Color p, sf::Vector2u size, LFSR
1)
104: {
105:     for (unsigned int x = 0; x<size.x; x++)
106:     {
107:         for (unsigned int y = 0; y< size.y; y++)
108:         {
109:             int new_bit1, new_bit2, new_bit3;
110:
111:             new_bit1 = l.generate(8);
112:             new_bit2 = l.generate(8);
113:             new_bit3 = l.generate(8);
114:
115:             p = image.getPixel(x, y);
116:             p.r = p.r ^ new_bit1;
117:             p.g = p.g ^ new_bit2;
118:             p.b = p.b ^ new_bit3;
119:             image.setPixel(x, y, p);
120:         }
```

```
121:    }  
122: }
```