```
1: /*
    2: Name: Albara Mehene
    3: Date: 10/1/2016
    4: Computing IV
    5:
    6: */
    7:
    8: #include <SFML/System.hpp>
    9: #include <SFML/Window.hpp>
   10: #include <SFML/Graphics.hpp>
   11:
   12: #include "LFSR.hpp"
   13:
   14: sf::Image transform(sf::Image picture, LFSR lfsr);
   16: int main(int argc, char* argv[])
   17: {
   18:
               if(argc < 5){
   19:
   20:
                                std::cout << "input-file.png, output-file.png, seed,</pre>
 tap" << std::endl;
   21:
                                return -1;
   22:
               }
   23:
   24:
               std::string input = argv[1];
   25:
               std::string output = argv[2];
   26:
               std::string i_seed = argv[3];
   27:
               int i_tap = atoi(argv[4]);
   28:
   29:
               sf::Image image;
               if (!image.loadFromFile(input))
   30:
                       return -1;
   31:
   32:
   33:
               sf::Image image e = image;
   34:
               sf::Image temp_e;
   35:
   36:
               //Pass the seed and tap function
   37:
               LFSR l(i_seed, i_tap);
   38:
               temp_e = transform(image_e, 1);
   39:
   40:
   41:
               sf::Vector2u size = image.getSize();
               sf::Vector2u size2 = temp_e.getSize();
   42:
   43:
               sf::RenderWindow window(sf::VideoMode(size.x, size.y), "Picturel");
   44:
               sf::RenderWindow window1(sf::VideoMode(size2.x, size2.y), "Picture2"
);
   45:
   46:
   47:
   48:
   49:
               sf::Texture texture;
               texture.loadFromImage(image);
   50:
   51:
   52:
               sf::Texture texture_e;
   53:
               texture_e.loadFromImage(temp_e);
   54:
   55:
               sf::Sprite sprite;
   56:
               sprite.setTexture(texture);
   57:
   58:
               sf::Sprite sprite_e;
   59:
               sprite_e.setTexture(texture_e);
```

```
Sat Oct 01 15:15:20 2016
PhotoMagic.cpp
   60:
   61:
               while (window.isOpen() && window1.isOpen())
   62:
   63:
                        sf::Event event;
   64:
                        while (window.pollEvent(event))
   65:
   66:
                                if (event.type == sf::Event::Closed)
   67:
                                         window.close();
   68:
   69:
                        while (window1.pollEvent(event))
   70:
   71:
                                if (event.type == sf::Event::Closed)
   72:
                                         window1.close();
   73:
                        }
   74:
   75:
   76:
                        window.clear(sf::Color::White);
   77:
                        window1.clear(sf::Color::White);
   78:
                        window.draw(sprite);
   79:
                        window1.draw(sprite_e);
   80:
                        window.display();
   81:
                        window1.display();
   82:
                }
   83:
               // fredm: saving a PNG segfaults for me, though it does properly
   84:
   85:
               // write the file
   86:
               if (!temp_e.saveToFile(output))
   87:
                        return -1;
   88:
   89:
               return 0;
   90: }
   91:
   92: sf::Image transform(sf::Image picture, LFSR lfsr){
   93:
               // p is a pixel
   94:
               sf::Color p;
   95:
               int temp;
   96:
               sf::Vector2u size = picture.getSize();
   97:
   98:
               // create photographic negative image of upper-left 200 px square
   99:
               for (unsigned int x = 0; x < size.x; x++) {
  100:
                        for (unsigned int y = 0; y < size.y; y++) {
  101:
                                p = picture.getPixel(x, y);
  102:
  103:
                                temp = lfsr.generate(8);
  104:
                                p.r = p.r ^ temp;
  105:
  106:
                                temp = lfsr.generate(8);
  107:
                                p.g = p.g ^ temp;
  108:
  109:
                                temp = lfsr.generate(8);
  110:
                                p.b = p.b ^ temp;
  111:
  112:
                                picture.setPixel(x, y, p);
  113:
  114:
  115:
               return picture;
  116: }
  117:
  118:
  119:
```

120:

PhotoMagic.cpp Sat Oct 01 15:15:20 2016 3

121: