```
    1: /*************************************************************************
***********/
    2: /* pixels.cpp
          */
    3: /* Yoo Min Cha
   */
    4: /* Linear Feedback Shift Register
          */
    5: /* Professor Martin
   */
    6: /* 06 March 2014
   */
    7: /*************************************************************************
***********/
    8:
    9: #include <SFML/System.hpp>
   10: #include <SFML/Window.hpp>
   11: #include <SFML/Graphics.hpp>
   12: #include <string>
   13: #include <cstdlib>
   14: #include "LFSR.hpp"
   15:
   16: using namespace std;
   17: using namespace sf;
   18:
   19: int main(int argc, char *argv[])
   20: {
   21:     // input arguments
   22:   if(argc != 5)
   23:   {
   24:     cerr << "Need source image filename, output image filename, and LFSR
seed and tap position.\n";
   25:     return EXIT_FAILURE + 4;
   26:   }
   27:   string sourceFile = argv[1];
   28:   string outputFile = argv[2];
   29:   string seed = argv[3];
   30:   unsigned int tap = atoi(argv[4]);
   31:
   32:    // load image from file
   33:   Image image;
   34:   if (!image.loadFromFile(sourceFile))
   35:     return EXIT_FAILURE + 6;
   36:   Image image2;
   37:   if (!image2.loadFromFile(sourceFile))
   38:     return EXIT_FAILURE + 7;
   39:
   40:    // instantiate a new lfsr object
   41:   LFSR lfsr(seed, tap);
   42:
   43:    // p is a pixel
   44:   Color p;
   45:
   46:   Vector2u size = image.getSize();
   47:   RenderWindow window(VideoMode(size.x *2, size.y), "Image Before/After")
;
   48:
   49:     // tranform image using lfsr method generate
   50:   for (int x=0; x < size.x; x++) {
   51:     for (int y= 0; y < size.y; y++) {
   52:       p = image.getPixel(x, y);
   53:       p.r ^= lfsr.generate(tap);
   54:       p.g ^= lfsr.generate(tap);
   55:       p.b ^= lfsr.generate(tap);
   56:       image.setPixel(x, y, p);
```

```
   57:     }
   58:   }
   59:
   60:   Texture texture;
   61:   texture.loadFromImage(image);
   62:
   63:   Texture texture2;
   64:   texture2.loadFromImage(image2);
   65:
   66:   Sprite sprite;
   67:   sprite.setTexture(texture);
   68:   sprite.setPosition(size.x, 0);
   69:
   70:   Sprite sprite2;
   71:   sprite2.setTexture(texture2);
   72:
   73:   while (window.isOpen())
   74:   {
   75:     Event event;
   76:     while (window.pollEvent(event))
   77:     {
   78:       if (event.type == Event::Closed)
   79:         window.close();
   80:     }
   81:
   82:     window.clear(Color::White);
   83:     window.draw(sprite);
   84:     window.draw(sprite2);
   85:     window.display();
   86:   }
   87:
   88:    //   write to new file
   89:   if (!image.saveToFile(outputFile))
   90:     return EXIT_FAILURE + 8;
   91:
   92:   return EXIT_SUCCESS;
   93: }
```