

```
1: #include <iostream>
2: #include <SFML/Graphics.hpp>
3: #include "Triangle.hpp"
4:
5: using std::cout;
6: using std::endl;
7: using std::stod;
8: using sf::RenderWindow;
9: using std::stoi;
10:
11: void triangleFractal(int i, RenderWindow* window,
12:     double x, double y, double l) {
13:
14:     Triangle triangle(x, y, l);
15:     window->draw(triangle);
16:     if (i > 0) {
17:         triangleFractal(i - 1, window, x - (l / 4),
18:             y - sqrt((3.0/16) * pow(l, 2)), l/2);
19:         triangleFractal(i - 1, window, x + l, y, l/2);
20:         triangleFractal(i - 1, window, x, y + sqrt(.75 * pow(l, 2)), l/2)
;
21:     }
22:     return;
23: }
24:
25: int main(int argc, char* argv[]) {
26:     // sets up command line arguments
27:     if (argc != 3) {
28:         cout << "Incorrect number of inputs." << endl;
29:         exit(1);
30:     }
31:     int N = stoi(argv[2]);
32:     double L = stod(argv[1]);
33:     if (L <= 0 || N <= 0) {
34:         cout << "Incorrect input range." << endl;
35:     }
36:
37:     RenderWindow window(sf::VideoMode(L * 3, L * 3), "PS3: Recursive Grap
hics (Triangle Fractal)");
38:     // loop to check if closed
39:     while (window.isOpen()) {
40:         sf::Event event;
41:         while (window.pollEvent(event)) {
42:             if (event.type == sf::Event::Closed)
43:                 window.close();
44:         }
45:
46:         // fractal triangle setup
47:         window.clear(sf::Color::White);
48:         triangleFractal(N, &window, L * (9.0/10), L, L);
49:
50:         window.display();
51:     }
52:     return 0;
53: }
```