```
/****************************************************
 * Homework 1: Rasterization                        *
 * CS 148 (Summer 2016), Stanford University        *
 *--------------------------------------------------*
 * Here you will implement the circle rasterization *
 * method you derived in the written portion of the *
 * homework.                                        *
 * To compile this in linux:                        *
 *        g++ hw1.cpp                               *
 * Then, run the program as follows:                *
 *        ./a.out 200                               *
 * to generate a 200x200 image containing a circular*
 * arc.  Note that the coordinate system we're using*
 * places pixel centers at integer coordinates and  *
 * has the origin at the lower left.                *
 * Your code will generate a .ppm file containing the*
 * final image. These images can be viewed using    *
 * "display" in Linux or Irfanview in Mac/Windows.  *
 ****************************************************/

#include <iostream>
#include <fstream>
#include <cstdio>
#include <cassert>
using namespace std;

// We'll store image info as globals; not great programming practice
// but ok for this short program.
int size;
bool **image;

void renderPixel(int x, int y) {
        assert(x >= 0 && y >= 0 && x <= size && y <= size);
        image[x][y] = 1;

        // TODO:  light up the pixel's symmetric counterpart
}

void rasterizeArc(int radius) {
        // TODO:  rasterize the arc using renderPixel to light up pixels
}

// You shouldn't need to change anything below this point.

int main(int argc, char *argv[]) {
        if (argc != 2) {
                cout << "Usage: " << argv[0] << " circleSize\n";
                return 0;
        }

#ifdef _WIN32
        sscanf_s(argv[1], "%d", &size);
#else
        sscanf(argv[1], "%d", &size);
#endif
        if (size <= 0) {
                cout << "Image must be of positive size.\n";
                return 0;
        }

        // reserve image as 2d array
        image = new bool*[size+1];
        for (int i = 0; i <= size; i++) image[i] = new bool[size+1];
```

```
        rasterizeArc(size);

        char filename[50];
#ifdef _WIN32
        sprintf_s(filename, 50, "circle%d.ppm", size);
#else
        sprintf(filename, "circle%d.ppm", size);
#endif

        ofstream outfile(filename);
        outfile << "P3\n# " << filename << "\n";
        outfile << size+1 << ' ' << size+1 << ' ' << 1 << endl;

        for (int i = 0; i <= size; i++)
        for (int j = 0; j <= size; j++)
                outfile << image[size-i][j] << " 0 0\n";

        // delete image data
        for (int i = 0; i <= size; i++) delete [] image[i];
        delete [] image;

        return 0;
}
```