Arnik Shah

Professor Wong

Artificial Intelligence

November 9, 2024

Project Report

Before you compile my program, in the main() function of the source code, manually enter the names of the input file (e.g., Input1.txt) and output file (e.g., Output1.txt).

To compile and run the C++ robot path planning program on Windows, install the g++ compiler by downloading and installing MinGW from the official website. Make sure you are using g++ version 4.8.1 or later and that your system supports C++11 or newer. You can check your version by typing **g++ --version** in the command prompt. Then, open the command prompt (cmd) and use the cd command to navigate to the directory where my C++ source file (AStarSearch.cpp) and the input files are located. In the command prompt, compile the program with the command **g++ -o AStarSearch AStarSearch.cpp**, which will generate an executable file named AStarSearch.exe. Once compiled, run the program by typing **./AStarSearch** in the command line, and it will prompt you to input a value for k, which must be an integer.

After you enter the value of k, the program will then attempt to find a path from the start to the goal and save the results in an output file. It will print whether a solution was found or not in the console and will put details about the nodes generated, depth of goal node, actions, f(n) values, and workspace in the output file.

Source Code:

```
// Arnik Shah
// Robot Path Planning Project
// November 9, 2024
```

```
#include <iostream>
#include <fstream>
#include <vector>
#include <queue>
#include <cmath>
#include <map>
#include <utility>
#include <iomanip> // For formatting output

using namespace std;

// Global Constants
const int rows = 30;
const int cols = 50;
const vector<pair<int, int>> directions = {
            {1, 0},    // Right (0 degrees)
            {1, 1},    // Up-Right (45 degrees)
            {0, 1},    // Up (90 degrees)
            {-1, 1},   // Up-Left (135 degrees)
            {-1, 0},   // Left (180 degrees)
            {-1, -1},  // Down-Left (225 degrees)
            {0, -1},   // Down (270 degrees)
            {1, -1},   // Down-Right (315 degrees)
        };
const vector<int> angles = {0, 45, 90, 135, 180, 225, 270, 315};

// Function Prototype for calculating Euclidean distance
double euclideanDistance(pair<int, int> a, pair<int, int> b);

// Class to calculate path cost (f(n) values) for each node
class NodePathCost {
public:
    NodePathCost(int k, int g) : k(k), g(g), h(0) {}

    void updateCost(int currentAngle, int newAngle, int action) {
        // Calculate Angle cost
        int angleDiff = abs(newAngle - currentAngle);
        if (angleDiff > 180) {
```

```
            angleDiff = 360 - angleDiff;
        }
        double angleCost = k * ((double)angleDiff / 180);
        // Calculate Distance cost
        double distanceCost = (action % 2 == 0) ? 1 : sqrt(2);
        // Update g(n) value
        g += angleCost + distanceCost;
    }

    void setHeuristicCost(pair<int, int> currPos, pair<int, int> goalPos) {
        // Get Euclidian distance between the current position and the goal position
        h = euclideanDistance(currPos, goalPos);
    }

    double getTotalCost() const { return g + h; }

    double getG() const { return g; }

private:
    int k;
    double g;
    double h;
};

// Class for each state
class Node {
public:
    Node(pair<int, int> loc, int angle, int k, int g) : location(loc), angle(angle),
cost(k, g) {}

    void updateNodePathCost(int action, int newAngle, pair<int, int> goalPos) {
        cost.updateCost(angle, newAngle, action);
        cost.setHeuristicCost(location, goalPos);
        angle = newAngle;
    }

    double getCost() const { return cost.getTotalCost(); }

    double getTraveledcost() const { return cost.getG(); }
```

```
    pair<int, int> getLocation() const { return location; }

    int getAngle() const { return angle; }

    const vector<int>& getPrevActions() const { return prevActions; }

    const vector<double>& getPrevCosts() const { return prevfValues; }

    void setPrevActions(const vector<int>& actions) { prevActions = actions; }

    void setPrevCosts(const vector<double>& costs) { prevfValues = costs; }

    void addActions(int action) { prevActions.push_back(action); }

    void addCosts(double cost) { prevfValues.push_back(cost); }

    // Used to compare nodes in the frontier
    bool operator<(const Node& other) const { return this->getCost() >
other.getCost(); }

    bool operator==(const Node& other) const { return location == other.location; }
private:
    pair<int, int> location;
    int angle;
    NodePathCost cost;
    // Stores the path actions taken to reach this node
    vector<int> prevActions;
    // Stores f(n) values along the path
    vector<double> prevfValues;
};

// A* Search Class
class AStarSearch {
public:
    AStarSearch(pair<int, int> start, pair<int, int> goal, int** workspace, int k)
        : start(start), goal(goal), workspace(workspace), k(k) {}

    ~AStarSearch() {
```

```
        // Free each row
        for (int c = 0; c < cols; ++c) {
            delete[] workspace[c];
        }
        // Free array of pointers
        delete[] workspace;
    }

    bool solve() {
        priority_queue<Node> frontier;
        map<pair<int, int>, double> visited;
        // Add initial state to frontier and set
        Node startNode(start, 0, k, 0);
        frontier.push(startNode);
        visited[start] = startNode.getCost();

        while (!frontier.empty()) {
            Node current = frontier.top();
            frontier.pop();

            // Check if the current node is the goal
            if (current.getLocation() == goal) {
                solutionActions = current.getPrevActions();
                solutionFValues = current.getPrevCosts();
                solutionFValues.push_back(current.getCost()); // Include f(n) at the
goal
                return true;
            }
            // Expand Node if current is not the goal
            // Check all possible actions
            for (size_t i = 0; i < directions.size(); ++i) {
                pair<int, int> newLoc = {
                    current.getLocation().first + directions[i].first,
                    current.getLocation().second + directions[i].second
                };
                // Check if location is in bounds and if it does not have an obstacle
                if (inBounds(newLoc) && workspace[newLoc.first][newLoc.second] != 1)
{
                    // Make new Node and update its f(n) value
```

```
                    Node newNode(newLoc, current.getAngle(), k,
current.getTraveledcost());
                    newNode.updateNodePathCost(i, angles[i], goal);

                    // Copy prev actions and prev f(n) values from the current node
                    newNode.setPrevActions(current.getPrevActions());
                    newNode.setPrevCosts(current.getPrevCosts());
                    newNode.addActions(i);
                    newNode.addCosts(newNode.getCost());
                    // Check if the new node is not in the map or has a smaller cost
than one in the map
                    if (visited.find(newLoc) == visited.end() || newNode.getCost() <
visited[newLoc]) {
                        frontier.push(newNode);
                        visited[newLoc] = newNode.getCost();
                        nodesGenerated++;
                    }
                }
            }
        }
        return false;
    }

    const vector<int>& getSolutionActions() const { return solutionActions; }

    const vector<double>& getSolutionFValues() const { return solutionFValues; }

    int getNodesGenerated() const { return nodesGenerated; }

private:
    pair<int, int> start, goal;
    int** workspace;
    int k;
    int nodesGenerated = 1;

    vector<int> solutionActions;
    vector<double> solutionFValues;

    bool inBounds(pair<int, int> loc) const {
```

```
        return loc.first >= 0 && loc.first < cols && loc.second >= 0 && loc.second <
rows;
    }
};

// Function prototypes
pair<int, int> getPosition(ifstream& input);
int** getWorkspace(ifstream& input);
void outputResults(const string& filename, AStarSearch& search, int** currentState,
pair<int, int> start);

int main() {
    ifstream input("Input1.txt");
    if (!input) {
        cerr << "Could not open the file.\n";
        exit(1);
    }

    pair<int, int> start = getPosition(input);
    pair<int, int> goal = getPosition(input);
    int** workspace = getWorkspace(input);
    input.close();

    // Get k value
    int k;
    cout << "What is the value of k?" << endl;
    cin >> k;
    AStarSearch search(start, goal, workspace, k);
    bool findSoultion = search.solve();

    // Output the nodes generated, solution actions, solution costs, and solution
workspace
    if (findSoultion) {
        outputResults("Output.txt", search, workspace, start);
        cout << "Solution found and saved to output file." << endl;
    } else {
        cout << "No solution found." << endl;
    }
}
```

```
double euclideanDistance(pair<int, int> a, pair<int, int> b) {
    return sqrt(pow(b.first - a.first, 2) + pow(b.second - a.second, 2));
}

pair<int, int> getPosition(ifstream& input) {
    // Get starting and final coordinates
    int iCoordinate, jCoordinate = 0;
    input >> iCoordinate >> jCoordinate;
    return {iCoordinate, jCoordinate};
}

int** getWorkspace(ifstream& input) {
    // Allocate 50 columns
    int** matrix = new int*[cols];
    for (int c = 0; c < cols; ++c) {
        // For each column, allocate 30 rows
        matrix[c] = new int[rows]();
    }

    // Read the workspace values
    for (int j = rows - 1; j >= 0 ; --j) {
        for (int i = 0; i < cols; ++i) {
            input >> matrix[i][j];
        }
    }
    return matrix; // Return the raw pointer
}


void outputResults(const string& filename, AStarSearch& search, int** workspace,
pair<int, int> start) {
    ofstream output(filename);
    // Depth of Node
    output << search.getSolutionActions().size() << endl;
    // Total nodes generated
    output << search.getNodesGenerated() << endl;

    // Solution path (actions)
```

```
    for (int action : search.getSolutionActions()) {
        output << action << " ";
    }
    output << endl;

    // f(n) values along the solution path
    for (double fValue : search.getSolutionFValues()) {
        // Show all values with 2 decimal points
        output << fixed << setprecision(2) << fValue << " ";
    }
    output << endl;

    // Mark the path in the currentState
    for (size_t i = 1; i < search.getSolutionActions().size(); ++i) {
        int action = search.getSolutionActions()[i];
        start.first += directions[action].first;
        start.second += directions[action].second;
        workspace[start.first][start.second] = 4;
    }

    // Print Workspace
    for (int row = rows - 1; row >= 0; --row) {
        for (int col = 0; col < cols; ++col) {
            output << workspace[col][row] << " ";
        }
        output << endl;
    }

    output.close();
}
```

Output Files:

Output for Input1.txt and k = 0:

```
31
134
7 0 0 7 7 7 7 7 7 7 7 7 0 7 7 0 0 0 0 0 0 0 0 1 1 1 0 0 0 7 0
```

```
32.74 32.36 32.41 32.57 32.34 32.12 31.93 31.76 31.62 31.51 31.44 31.41 31.00 31.44
31.54 31.13 31.14 31.15 31.17 31.18 31.20 31.22 31.25 31.49 31.41 31.51 31.12 31.16
31.24 31.41 31.00 31.00
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 2 4 4 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 1 0 0 0 0 4 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 4 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 4 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 4 0 0 0 0 1 0 0 0 0 0 0 0 0 1 4 4 4 4 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 1 1 4 4 0 0 1 1 1 0 0 1 1 0 0 4 0 0 0 0 4 4 0 1 1 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0 4 0 0 1 0 0 0 1 1 1 4 0 0 0 0 1 1 0 5 1 1 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 4 4 4 4 4 4 4 4 4 1 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

Output for Input1.txt and k = 2:

```
31
159
7 0 0 7 7 7 7 7 7 7 7 7 0 7 7 0 0 0 0 0 0 0 0 1 1 1 0 0 0 7 0
33.24 32.86 32.41 33.07 32.34 32.12 31.93 31.76 31.62 31.51 31.44 31.41 31.50 31.94
31.54 31.63 31.14 31.15 31.17 31.18 31.20 31.22 31.25 31.99 31.41 31.51 31.62 31.16
31.24 31.91 31.50 31.50
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 2 4 4 0 1 1 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0
0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 4 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 4 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 4 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 4 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 4 4 4 4 1 1 1 1 0
0 0 0 0 0 0 0

```
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 1 1 4 4 0 0 1 1 1 0 0 1 1 0 0 4 0 0 0 0 4 4 0 1 1 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 1 1 0 0 4 0 0 1 0 0 0 1 1 1 4 0 0 0 0 1 1 0 5 1 1 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 4 4 4 4 4 4 4 4 4 1 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

Output for Input1.txt and k = 4:

```
31
330
0 0 7 7 7 7 7 7 7 7 7 7 0 7 7 7 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1
32.62 32.68 33.83 33.57 33.34 33.12 32.93 32.76 32.62 32.51 32.44 32.41 33.00 34.44
34.54 34.71 35.32 35.34 35.37 35.40 35.44 35.49 35.54 35.62 35.71 35.83 36.00 36.24
37.24 36.83 36.41 36.41
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 2 4 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 4 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 4 0 0 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 0 0 4 0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 0 0 0 4 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 1 1 0 1 1 0 0 4 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 4 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 1 4 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 0 1 1 4 4 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 4 5 1 1 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 0 0 0 1 1 0 0 1 4 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 4 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 4 0 0 0 0 0 0 0 0 0 1 1 0 0 4 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 4 4 4 4 4 4 4 4 4 4 4 4 4 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

Output for Input2.txt and k = 0:

```
37
318
7 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 2 2 2 1 0 0 0 1 1 1 2 2 2 1 1 2 1 1 0 0
39.06 38.88 39.12 39.38 39.66 39.66 39.24 38.83 38.41 38.00 37.59 37.46 37.76 38.08
38.09 37.67 37.26 37.10 37.38 37.69 37.69 37.56 37.87 38.21 38.22 37.82 37.41 37.22
37.49 37.81 37.82 37.41 37.47 37.58 37.41 37.00 37.00 37.00
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 4 4 4 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 4 0 0 0 5 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 4 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 4 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 4 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 4 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 4 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 4 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 4 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 4 4 4 4 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 4 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 1 4 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 4 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 4 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 4 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 4 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 4 4 4 4 1 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 4 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 4 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 4 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 4 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 4 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 2 4 4 4 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

Output for Input2.txt and k = 2:

```
37
352
7 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 2 2 2 1 0 0 0 1 1 1 2 2 2 1 1 2 1 1 0 0
39.56 39.38 39.12 39.38 39.66 40.16 39.24 38.83 38.41 38.00 37.59 37.96 37.76 38.08
38.59 37.67 37.26 37.60 37.38 37.69 38.19 38.06 37.87 38.21 38.72 37.82 37.41 37.72
37.49 37.81 38.32 37.41 37.97 38.08 37.41 37.50 37.00 37.00
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 4 4 4 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 4 0 0 0 5 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 4 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 4 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 4 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 4 0 1 1 0 0 1 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 4 0 0 1 1 0 0 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 4 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 4 0 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 4 4 4 4 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 4 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 1 4 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0
```

```
0 0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 4 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 1 0 1 4 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 4 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 4 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 4 4 4 4 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 4 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 4 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 4 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 4 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 4 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 2 4 4 4 4 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

Output for Input2.txt and k = 4:

```
37
575
0 0 7 0 0 1 1 1 1 1 1 0 0 0 1 1 1 2 2 2 1 0 0 0 1 1 1 2 2 2 2 1 1 1 1 0 0
38.01 38.24 40.54 41.38 41.66 42.66 42.24 41.83 41.41 41.00 40.59 41.46 41.76 42.08
43.09 42.67 42.26 43.10 43.38 43.69 44.69 45.56 45.87 46.21 47.22 46.82 46.41 47.22
47.49 47.81 48.21 49.25 48.89 48.58 48.41 49.00 49.00 49.00
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 1 4 4 4 5 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 4 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 4 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 4 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 4 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 4 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 4 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 4 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 4 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 0 0 0 0 0 0 0 0 0 4 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 4 4 4 4 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 1 0 0 1 4 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 0 0 0 1 1 0 0 1 4 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 0 0 0 1 1 1 0 1 4 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 4 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 4 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 1 1 0 0 0 0 1 4 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 0 4 4 4 4 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 1 1 4 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 0 4 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 1 0 0 0
0 0 0 0 0 0 0
0 0 0 1 1 0 1 1 1 0 4 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 4 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 2 4 0 1 1 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 4 4 4 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
```

Output for Input3.txt and k = 0:

```
48
236
1 1 2 2 2 2 2 1 0 1 0 1 1 1 2 1 0 7 0 0 0 0 1 1 1 0 7 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1
0 0 1 0 0 0
46.60 46.28 46.46 47.08 47.72 48.38 49.06 49.22 48.85 49.01 48.64 48.81 48.59 48.37
48.73 48.97 48.58 49.22 48.83 48.86 48.89 48.93 49.12 48.93 48.76 48.36 49.00 48.62
48.65 48.85 48.46 48.49 48.52 48.56 48.60 48.65 48.70 48.77 48.85 48.96 48.69 48.50
48.10 48.12 48.41 48.00 48.00 48.00 48.00
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 5 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 4
4 4 4 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 4 4 4 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 4 0 0 0 0
0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0
0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 4 4 1 0 0 4 4 4 4 4 4 4 4 4 4 0 0 0 0 0 0
0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 4 0 1 4 4 4 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 4 4 1 0 0 0 0 4 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 4 0 0 4 4 4 4 4 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0
1 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0
1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 4 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 4 4 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 0 4 4 1 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 1 4 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0

```
0 1 1 1 0 0 0 0
0 0 0 2 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

Output for Input3.txt and k = 2:

```
48
229
1 1 2 2 2 2 2 1 0 1 0 1 1 1 2 1 1 0 7 7 0 0 1 1 1 0 7 0 0 1 0 0 0 0 0 0 0 0 0 1 1 1
1 0 0 0 0 0
47.10 46.28 46.96 47.08 47.72 48.38 49.06 49.72 49.35 49.51 49.14 49.31 48.59 48.37
49.23 49.47 48.81 48.91 49.53 49.28 49.39 48.93 49.62 48.93 48.76 48.86 49.50 49.12
48.65 49.35 48.96 48.49 48.52 48.56 48.60 48.65 48.70 48.77 48.85 49.46 48.69 48.50
48.41 48.50 48.00 48.00 48.00 48.00 48.00
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 5 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 4 4 4
4 4 4 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 4 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 4 0 0 0 0
0 0 1 1 1 0 0 0
```

0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0
0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 4 4 1 0 0 4 4 4 4 4 4 4 4 4 4 0 0 0 0 0 0
0 0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 0 4 4 0 0 0 1 1 4 0 1 4 4 4 0 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 1 4 1 1 4 0 0 0 4 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 4 0 0 1 0 4 4 4 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0
1 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 0 4 0 1 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0
1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 1 4 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 1 0 4 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 4 4 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 0 4 4 1 1 0 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 1 4 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 1 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 2 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```
0 0 0 0 0 0 0 0
```

Output for Input3.txt and k = 4:

```
48
596
1 1 2 2 2 2 2 1 0 0 1 1 1 1 2 1 0 0 0 0 7 7 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 1 1 1
1 0 0 0 0 0
47.60 47.28 48.46 49.08 49.72 50.38 51.06 52.22 52.85 52.90 54.06 53.81 53.59 53.37
54.73 55.97 56.58 56.59 56.61 56.64 58.31 58.62 59.25 59.30 59.35 59.41 59.47 60.60
60.34 60.10 59.88 60.49 60.52 60.56 60.60 60.65 60.70 60.77 60.85 61.96 61.69 61.50
61.41 62.00 62.00 62.00 62.00 62.00 62.00
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 1 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
1 1 1 1 1 1 1 1
0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 0 0 0 1 1 0 0 1 1 0 0 1 1 1 1 0 0 0 0 0 0 0 0 0 0
0 0 0 5 0 0 0 0
1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 0 0 0 1 1 0 0 0 0 0 0 1 1 1 1 0 0 0 0 4 4 4 4 4 4
4 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 0 0 0 1 1 1 1 0 1 1 0 0 1 1 1 1 0 0 4 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 1 1 4 0 0 0 0 0
0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 0 0 0 1 1 0 1 1 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 4 0 0 0 0 0 0
0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 1 1 0 0 0 1 0 0 0 4 4 4 4 4 4 4 4 4 0 0 0 0 0 0 0
0 1 1 1 0 0 0
0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 0 0 1 1 1 0 4 1 0 0 0 0 1 1 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 1 1 4 4 4 4 4 0 0 0 0 0 0 0 1 4 1 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0
```

```
0 0 0 0 0 0 0 0 0 0 0 4 0 0 1 0 0 4 0 0 0 0 0 0 4 0 0 0 0 1 1 1 1 0 0 1 1 1 0 0 0 0
1 1 0 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 0 4 0 1 1 1 0 0 4 4 4 4 4 4 1 1 0 0 0 1 0 1 1 0 0 0 1 1 0 0 0 0
1 1 1 0 0 0 0 0
0 0 0 1 1 1 1 1 1 1 4 0 0 0 1 1 0 0 1 1 0 1 1 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 4 0 0 0 0 1 1 1 0 1 0 0 1 1 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 4 0 0 0 0 0 0 1 1 0 0 0 0 0 0 1 1 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 0 4 4 4 1 0 0 0 0 0 0 1 1 1 0 0 1 1 0 0 1 1 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 1 1 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0 0 1 1 1 0 0 1 1 1 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 1 4 0 1 1 1 0 1 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 1 0 0 0 1 1 1 1 0 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 1 1 0 0 0 1 1 0 0 0 0 1 1 1 0 0 1 1 0 0 0 0 0 0 0 0 1 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 0 0 0 0 0 1 1 0 0 0 0 0 1 0 0 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 0
0 0 0 0 0 0 0 0
0 0 0 1 4 0 1 1 1 0 0 0 0 0 1 1 0 0 1 1 0 1 1 1 1 0 0 1 1 1 0 0 0 0 1 1 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 4 0 1 1 1 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0 0 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 2 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 1 1 1 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```