CPSC 406 – Homework 5
Best-First Robot Navigation
Due 4-16@7pm.

Problem Description

In this assignment, you will implement a solution to a navigation problem, which is as follows:

> A robot represented as a point moves in a regular two-dimensional NxN grid (e.g., N = 100). Each point of the grid is either "free" or "forbidden" (obstacle). From any position (i,j) in the grid the robot can reach each of the 4 adjacent positions (i,j-1), (i,j+1), (i-1,j), (i+1,j), if it is not forbidden. A navigation problem consists of finding a path in the grid (sequence of adjacent free points) that connects a given initial position to a given goal position.

General Information

Write a program implementing the Greedy Heuristic Search algorithm to solve this problem. To do this, first formulate the navigation problem as a search problem: what are the states, the successor function, the initial and goal states? Allowing the search tree to contain nodes labeled by the same state leads to an infinite search tree. So, the program should avoid duplicate states.

You should have sections in your program to test each of the following evaluation functions:

- f(N) = Euclidean distance from N to the goal (Strategy 1)
  (i.e. the length of a straight line between two points, E((i,j),(i',j')) = sqrt[(i-i')^2+(j-j')^2])

- f(N) = Manhattan distance to the goal (Strategy 2)
  (i.e. the length of the shortest path obtainable by traversing only in the cardinal directions, ignoring any obstacles, M((i,j),(i',j')) = |i-i'| + |j-j'|)

For each problem-function pair, your program should output the generated path, its cost, and the number of nodes in the search tree when the solution was found.

Getting Started

Your program will take as a command line argument an input file specifying the layout of the robot's environment.

The input files will contain information about the map that the robot will traverse. The first line will have a number N that is the width and the height of the map (all maps are NxN, i.e. same height and width). The following N lines will detail the map. Each line will have N characters, representing the N spaces in that row. The first line will be row N and the first character in each row will be in column 0. The characters in the rows will be as follows:

.	empty space (traversable)
i	initial space (traversable, robot start here)
g	goal space (traversable, robot's goal here)
+	obstacle (not traversable)

A sample input file might be as follows:

```
5
...g.
.++..
.i+..
..+..
+...+
```

Every input file is expected to have at least one possible path to the goal. Your robot cannot move diagonally; it can only move to the spaces immediately up, down, left, and right. The robot cannot enter a space occupied by an obstacle.

The maps that you will generate as output should be the same as the maps in the input files, except that each step in the path taken will be represented as an 'o' (the lower case letter o). The goal and initial spaces should remain 'g' and 'i' respectively. As an example, a solution map to the above input map might be:

```
ooog.
o++..
oi+..
..+..
+...+
```

Deliverables

Submit your commented source code, a README, and a Makefile as a tgz archive to the digital dropbox by the due date above. You will also demo your code in class.