Day 2 Task 3: Maze

In southern Ontario, many corn farmers create cornstalk mazes like the one shown. The mazes are created in the fall, after the grain has been harvested. There is still time for you to help design the best maze ever for 2010.

A field is covered with corn stalks except for a few obstacles (trees, buildings and the like) where corn cannot grow. The stalks, which are extremely tall, form the walls of the maze. Pathways are created on a square grid by crushing 1m square areas of stalks. One grid square on the edge is the entrance, and one grid square is the core of the maze.

Jack visits a corn maze every year, and has become adept at finding his way quickly from the entrance to the core. You are designing a new maze, and your job is to determine which stalks to crush, so as to maximize the number of squares Jack must visit.

The grader will determine which square is the entrance (the only one on the perimeter) and which square is the core (the one that Jack must walk farthest to reach).

A map of the rectangular field is represented as text; for example, a 6m by 10m field with eight trees might be represented as:

```
##X#######
###X######
####X##X##
##########
##XXXX####
##########
```

The symbol # represents a square with standing cornstalks, and X represents a square with an obstacle (such as a tree) that cannot be crushed to form a pathway.

The field is transformed into a maze by crushing squares occupied by corn. One crushed square (the entrance) must be on the edge of the field. The other crushed squares must be in the interior. The objective is to maximize the shortest path from the entrance to the core, measured by the number of crushed squares that Jack must pass through, including the entrance and the core. It is possible to pass from one square to another only if both are crushed and they share an edge.

In your submission, the crushed squares should be identified by periods (.). Exactly one of the crushed squares should be on the perimeter. For example:

```
#.X#######
#.#X#...##
#...X#.X.#
#.#......#
#.XXXX##.#
##########
```

Below, for illustration purposes only, we mark the entrance `E`, the core `C` and remainder of the path using `+`. The path length is 12.

```
#EX#######
#+#X#C+.##
#+++X#+X.#
```

```
#.#++++..#
#.XXXX##.#
##########
```

The folder `/home/ioi2010-contestant/maze` contains several text files named `field1.txt` `field2.txt` etc. containing maps of cornfields. You are to copy them to files named `maze1.txt` `maze2.txt` etc., and transform them into valid mazes by replacing some of the # symbols by periods.

Note: the Grading Server Public Test will award 1 point per subtask for any valid solution (regardless of the path length). The Grading Server Release Test will award the remaining points. The total score for the task will be rounded to the nearest integer between 0 and 110.

Subtask 1 [up to 11 points]

The field described above (of size 6x10) may be found in the file `field1.txt`. Create a maze for this field named `maze1.txt` that has a shortest path from the entrance to the core with length P. Your score for this subtask will be the minimum of 11 and $10^{P/20}$. Note that the sample solution scores 3.98 points.

Subtask 2 [up to 11 points]

The file `field2.txt` represents a field of size 100x100. Create a maze for this field named `maze2.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/4000}$.

Subtask 3 [up to 11 points]

The file `field3.txt` represents a field of size 100x100. Create a maze for this field named `maze3.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/4000}$.

Subtask 4 [up to 11 points]

The file `field4.txt` represents a field of size 100x100. Create a maze for this field named `maze4.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/4000}$.

Subtask 5 [up to 11 points]

The file `field5.txt` represents a field of size 100x100. Create a maze for this field named `maze5.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/5000}$.

Subtask 6 [up to 11 points]

The file `field6.txt` represents a field of size 11x11. Create a maze for this field named `maze6.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/54}$.

Subtask 7 [up to 11 points]

The file `field7.txt` represents a field of size 20x20. Create a maze for this field named `maze7.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/33}$.

Subtask 8 [up to 11 points]

The file `field8.txt` represents a field of size 20x20. Create a maze for this field named `maze8.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/95}$.

Subtask 9 [up to 11 points]

The file `field9.txt` represents a field of size 11x21. Create a maze for this field named `maze9.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/104}$.

Subtask 10 [up to 11 points]

The file `fieldA.txt` represents a field of size 200x200. Create a maze for this field named `mazeA.txt` that has a shortest path from the entrance to the core of length P. Your score for this subtask will be the minimum of 11 and $10^{P/7800}$.

Implementation Details

- This is an *output-only* task.
- Implementation folder: `/home/ioi2010-contestant/maze/` (prototype: maze.zip)
- To be submitted by contestant: `maze1.txt` `maze2.txt` `maze3.txt` `maze4.txt` `maze5.txt` `maze6.txt` `maze7.txt` `maze8.txt` `maze9.txt` `mazeA.txt`.
- Contestant interface: *none*
- Grader interface: *none*
- Sample grader: `grader.c` or `grader.cpp` or `grader.pas`
- Sample grader input: `grader.in.1` `grader.in.2` etc.
  Note: the implementation folder contains very simple solutions maze1.txt, maze2.txt etc.. Copy these to `grader.in.1` `grader.in.2` etc. for testing.
- Expected output for sample grader input: if the input is a valid maze for subtask *N*, the sample grader will output `OK N P` where `P` is the path length.
- Compile and run (command line): `runc grader.c` or `runc grader.cpp` or `runc grader.pas`
- Compile and run (gedit plugin): *Control-R*, while editing the grader.
- Submit (command line): `submit maze1.txt` or `submit maze2.txt` etc. *All .txt files in the implementation folder will be submitted, regardless of which is specified in the submit command.*
- Submit (gedit plugin): *Control-J*, while editing any .txt file.