

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 6×10) 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 100×100 . 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 100×100 . 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/`
- 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.