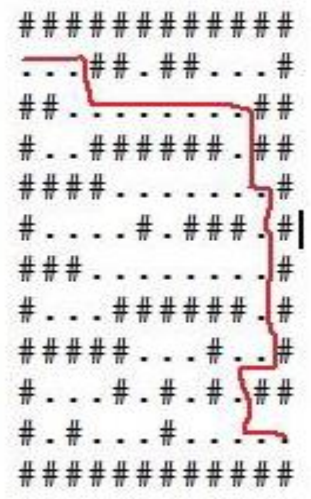
The following is a two-dimensional character array of size 12X12 which represents a maze. The symbol '#' represent a wall of the maze and the symbol '.' represents a possible path. There is an entry point and exit point of the maze, which are denoted by a '.' In the first column and 12th column respectively. Write a C program which takes text file which contains the maze as command line argument and prints the minimum number of moves required to go from entry to exit. If there is no path from entry to exit display "-1".

You need to formulate the maze as a graph and use Breadth First Search to find shortest path from entry to exit. You can treat each cell as a vertex and if we can go from one cell to other cell with one move add edge between the corresponding vertices.

For example, for the below maze:

```
############
...##.##...#
##........##
#..######.##
####.......#
#....#.###.#
###........#
#...######.#
#####...#..#
#...#.#.#.##
#.#...#.....
############
```

Minimum number of moves required is: 22



```
############
...##.##...#
##........##
#..######.##
####.......#
#....#.###.#
###........#
#...######.#
#####...#..#
#...#.#.#.##
#.#...#.....
############
```

Test cases attached should have the following output:
ptestcase1 : 22, ptestcase2 : 11, ptestcase3 : 19, ntestcase4 : -1