

CSCI2100B: Data Structures (Spring 2011)

Assigned: 11 Apr 2011

Version 1.0 (Updated: 10/04 12:48)

Due: 29 Apr 2011 23:59

Programming Assignment 4: Find me an Exit!

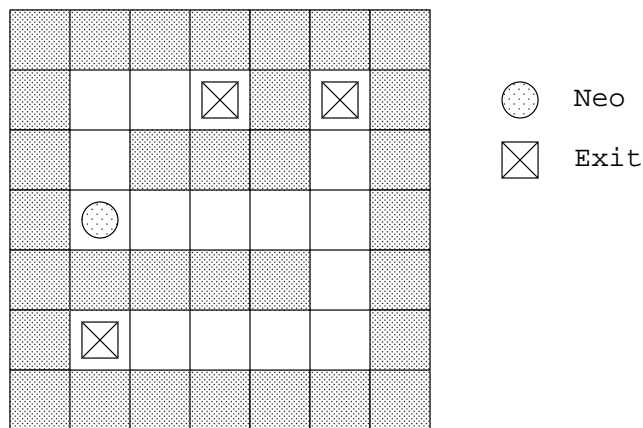
Limits: Runtime - 1 sec • Memory - 64 MB • Submission - 20 times

Background

Welcome to the Matrix.

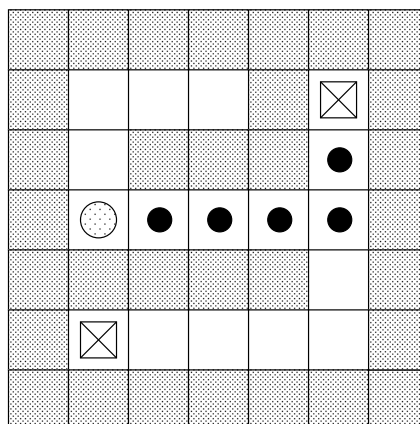
Neo has been trapped in the **Matrix**. He was chased by a group of Agents and urgently needed an exit to return to the real world. He called you, the **Operator**, to find him the nearest exit that can be reached from his current position.

Therefore you loaded the map of **Matrix** to your computer and started to search for an exit. The area where Neo was in can be considered as a 2D grid which is bounded by walls. And exits span across some grids in the area:



You spotted that it is not easy to find the nearest exit when the map becomes large. In addition, you have to identify the path to that nearest exit so that you can guide Neo through his escape from the Agents.

As a professional hacker, you decide to write a program to identify the exit and the corresponding path when the map is loaded. For example:



Problem

Write a program to find the path to the nearest exit accessible from Neo.

Input

The input begins with the dimension of the map: the first line contains the width (≤ 50) (the number of columns) and the second line contains the height (≤ 50) (the number of rows). Then it is followed by the **Matrix** map: each grid is represented by an ASCII character - # represents a wall; 0 represents Neo and X represents an exit. A walk-able grid is represented by a space.

You can assume that the map is always bounded by walls and there is only one Neo. But there can more than one exits within the whole area. Moreover, you may assume there is an *unique* path to every *accessible* exit and no two exits have the same distance (the number of grid steps) from Neo.

Sample Input:

```
7
7
#####
#    #X#
#   ##  #
#0    ##
#####
#X    ##
#####
```

Output

Mark the path to the nearest exit on the map using character . and output the whole map again. You do not have to print the dimensions.

Sample output:

```
#####
#    #X#
#   ##. #
#0...#
#####
#X    ##
#####
```

Test Cases & Scoring

Any reasonable (non-trivial, non-hard-coded) code that can be compiled successfully scores 2 marks.

Your program will be tested against a set of 8 test cases, namely:

1. a 5×5 map with 1 exit.
2. a 7×7 map with 2 exits. (Sample)
3. a 7×7 map with 1 exit.
4. a medium-size map with 3 exits.
5. Variant of map 4.
6. a medium-size map with many exits.
7. a large-size map with 3 exits.
8. a large-size map with many exits.

Every test case scores 1 mark.

Therefore your program can score 10 marks in total.

Submission

You should write your program in a single C source file. Submit your program using your UNIX account, following the instructions below:

1. Copy/upload it to your CSE UNIX account.
2. SSH to any CSE UNIX workstation through SSH, compress the source code by
`gtar zcvf <sid>.tar.gz <file_name>.c`
where <sid> is your 10-digit student ID and <file_name> is the name of the source code you wish to submit. Do **NOT** include other files.
3. Submit it to our Judge System by
`uuencode <sid>.tar.gz <sid>.tar.gz | mailx -s "ASG<code> <sid>" csci2100b`
where <sid> is your 10-digit student ID and <code> is the assignment code you are trying to solve.
4. Upon successful submission, you will receive a submission receipt and the judge reply very soon. You should **KEEP** your submission receipt for future references.

- END -