

ITCS 3146 – Spring 2016

Programming Assignment #1

The Pathfinder

(You may download this assignment statement from Moodle)

Deadline: Wednesday, February 10th, 2016, 11:59pm

Problem Statement

The following grid of ones and zeroes is a double scripted array representation of a terrain of size 12 x 12.

```
1 1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 0 0 0 0 1 0 1
0 0 1 0 1 0 0 0 0 1 0 1
1 1 0 0 1 0 0 0 0 1 0 1
1 0 0 0 0 0 1 1 0 1 0 0
1 1 1 1 0 0 1 1 0 1 0 1
1 0 1 1 0 0 1 1 0 1 0 1
1 1 1 1 0 0 1 1 0 1 0 1
1 0 0 0 0 0 0 0 0 1 0 1
1 1 1 1 1 0 0 0 0 1 0 1
1 0 0 0 0 0 0 0 0 1 0 1
1 1 1 1 1 1 1 1 1 1 1 1
```

The ones represent the obstacles in the field, and the zeroes represent positions in the possible path through the field.

In this assignment, the field will have a single entrance and a single exit, i.e., there will be only two zeroes in the “outer wall” of the field. In order to find the entrance and the exit, traverse the outer wall of the field in the clockwise direction, starting from the upper left corner. The first encountered zero will be the entrance (square [4,11] in the field above), and next zero will be the exit (square [2,0]). In this assignment, **the size of a field is NOT FIXED. Each array dimension will vary in size between 5 and 100.**

There are several simple algorithms for walking through a field that guarantee finding the path, if one exists. *The only legal moves are north, west, south or east (no diagonal moves)*. For example, look to your right and walk forward. Always keep the obstacle to your right. If you reach the corner of the obstacle, turn right and continue following its “border” on your right side. There may be a shorter path than the one you have taken, but in this way you are guaranteed to get out of the field. In this algorithm, if you exit from the field through the entrance, this means that the path from the entrance to the exit does not exist. Otherwise, the algorithm has found a path that avoids all obstacles.

In your assignment, you need to write a program called `path.c`. First of all, the program should ask user to type in the size of the field (*if you enter 12, 6, this will imply a 12 rows and 6 columns field*). Then the program should request the user to type in the name of the input file that contains the field (the code for reading the file is offered to you on Moodle). You may assume that the field size given to the program always matches the size in the file. The field will be given to your program in an ASCII text file, looking very much like the one above (but of proper size). Upon opening the file, your program must find the entrance. As your program attempts to find a path through the field, it should place the character `X` into each square visited in the path. Note that your program **MUST NOT** replace a `1` (an obstacle) with an `X`. Only zeroes can be replaced by `X`. Before exiting,

your program must display the field with the traversed path, that is, the path between the entrance and the exit (the path is marked by X). Your program must also report whether the path was found or not.

In case of the field from the figure above, the path could not be found and your program would provide the following output:

```
1 1 1 1 1 1 1 1 1 1 1
1 0 0 0 1 0 0 0 0 1 X 1
0 0 1 0 1 0 0 0 0 1 X 1
1 1 0 0 1 0 0 0 0 1 X 1
1 0 0 0 0 0 1 1 0 1 X X
1 1 1 1 0 0 1 1 0 1 X 1
1 0 1 1 0 0 1 1 0 1 X 1
1 1 1 1 0 0 1 1 0 1 X 1
1 0 0 0 0 0 0 0 0 1 X 1
1 1 1 1 1 0 0 0 0 1 X 1
1 0 0 0 0 0 0 0 0 1 X 1
1 1 1 1 1 1 1 1 1 1 1
```

No path found.

Submitting Your Assignment

In order to be graded, you must submit the following:

1. *Submit all the program files (source code) to Moodle. **Programs submitted through e-mail will be rejected! It is your responsibility to submit programs successfully. Do not submit your programs before Friday, February 6th.***
2. *Submit program documentation to Moodle together with the program by Wednesday, February 10th. This can be a Word file, PDF, or another commonly formatted text file. The documentation consists of a short description of your algorithm (not more than a page long), description of program's main data structures (Did you represent the field by an array? Are the elements of the field characters or integers, etc). If your program does not work properly or if it does not compile, you are requested to state this clearly in the documentation (what works, what doesn't).*
3. *Please note that Moodle keeps track of the submission time. Programs submitted AFTER February 10, 11:59PM will be penalized by 5% of the grade for each day of delay (including weekend days). No assignment will be accepted after 11:59PM, Sunday, February 14th.*
4. *Programs that cannot be compiled will receive up to 30% of the assignment grade. Only well documented, nicely structured, fully functional programs, will receive a full credit.*

Programs, written in C (C++), must compile using **gcc (g++)** compiler on koding.com. Programs that do not compile using **gcc (g++)** compiler may not receive credit. You are advised to keep a copy of your program in your own koding.com account (with the original time-stamp), just in case we need to check the time stamp.

Academic Honesty:

*Each student in the class is expected to develop his/her assignment alone. **Do not share programs, or program parts, with your colleagues. Violators of this policy will be held responsible for academic dishonesty, and will bear consequences in accordance to the rules and regulations of UNC Charlotte.***