

EE2310 Introduction to Programming - Final Project

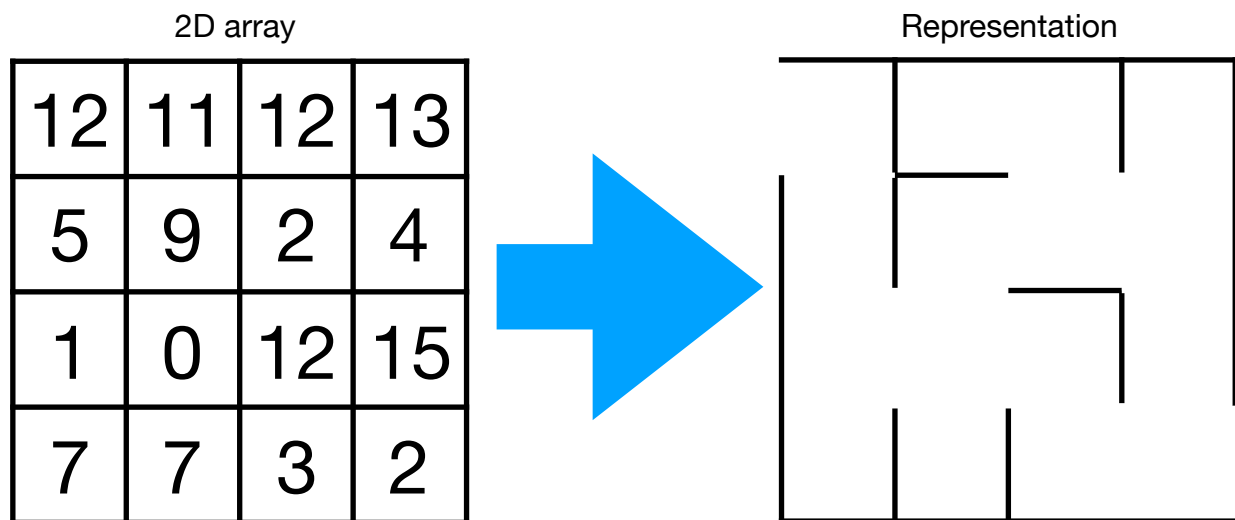
Due: **January 21, 2018 (11:59pm)**

Instructions

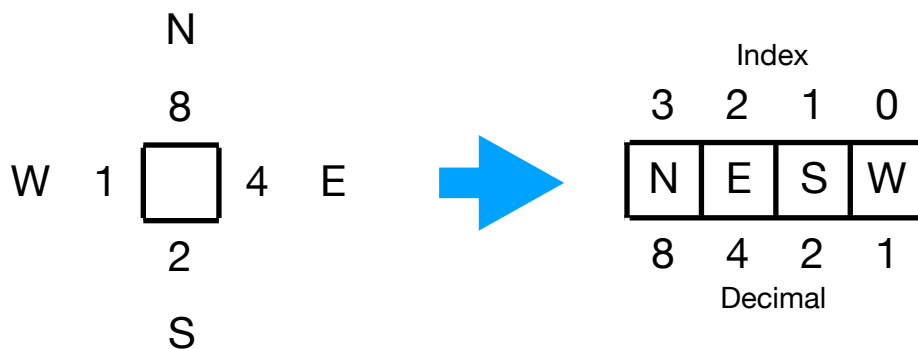
1. Submit your code on [NTHU online judge](#) and [iLMS system](#).
2. Name your source code as your student ID. For example, 103064533.cpp
3. Late submission will **incur 10% penalty per day up to 3 days**.

Rat in a Maze

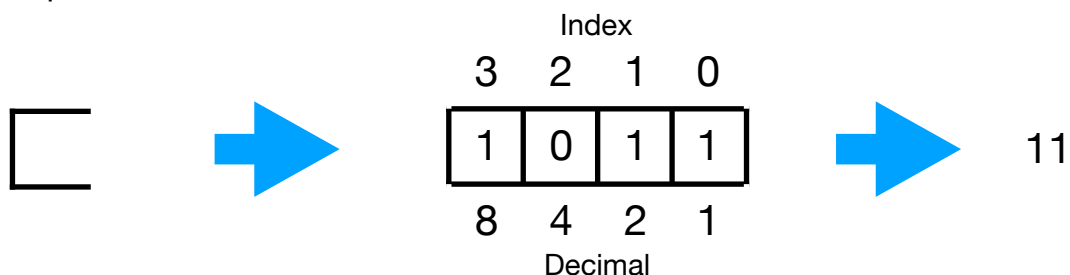
Rat in a maze is a popular problem that utilizes backtracking, linked list or stack. A maze is a 2D matrix in which some cells are blocked. The representation of a maze is given as the following 2D array:



We represent this maze a 4×4 array, where each element represents the corresponding “cell” in the following way. Each cell has 4 “doors”, and these doors can be either open or closed. All doors in the maze are “static”, i.e. they do not change over time. Each cell is represented as follows.



For example:



The four doors of a cell are represented as N, E, S, and W, respectively. We use a 4-bit value to represent the status of the doors. For example, (1011) in binary or 11 (since $8+2+1$) in decimal represents a cell with its E door opened and N, S, W doors are closed. That is, we use 0 represents the door opened and 1 represents the door closed. A 4×4 maze can be stored as a decimal-valued 4×4 array, where each corresponds to a number between 0 and 15. The W door of the top-left cell and E door of the bottom-right cell are always open, representing the entrance and exit of the maze. Note that the numbers of neighboring cells need to be consistent with each other.

What You Need to Do

Read in a $m \times n$ maze. There are 3 cases of output:

1. Output the shortest path from the entrance to the exit door. (There is only ONE shortest path.)
2. Output the inconsistent doors, such as sample 2.
3. If there is no path, print "There is no path."

In this assignment, you need to use C++ to finish the final project and you need to create a [structure](#) and a [class](#). Any kind of structure and class you can use and any place you can utilize the structure and the class.

NOTE: The maze may be some special cases, you need to think about those cases and avoid time limit exceeded or run time error.

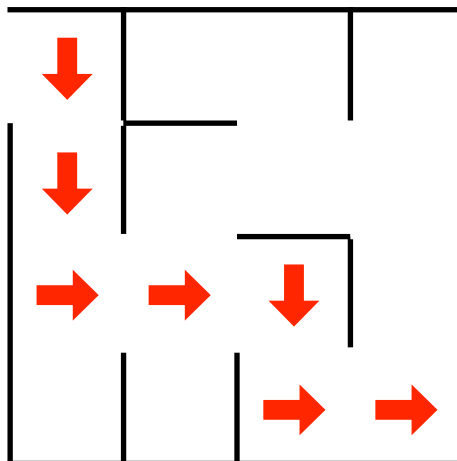
Input and Output Format

The input format: Two integers m and n , and a $m \times n$ array

The output format: See the sample output, remember to change a new line at the end.

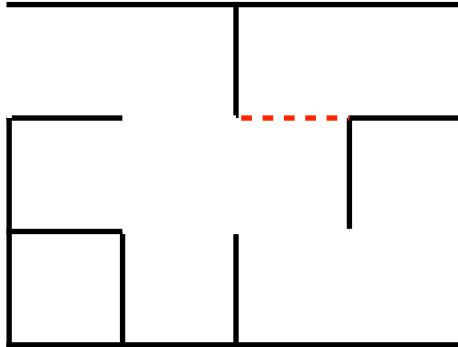
Sample Input and **output** 1: (Don't print any space in the path.)

```
4 4
12 11 12 13
5 9 2 4
1 0 12 5
7 7 3 2
(1,1)->(2,1)->(3,1)->(3,2)->(3,3)->(4,3)->(4,4)
```



Sample Input and **output 2**: (Put the smaller index of the cell in front of the other.)

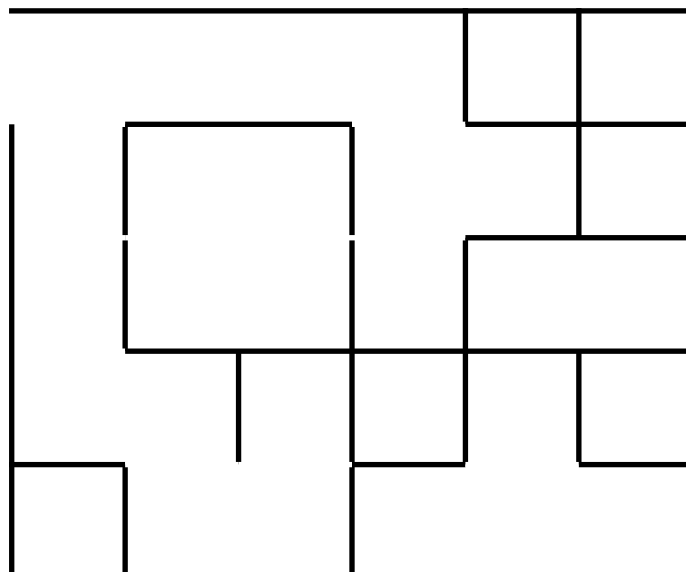
```
3 4
10 12 11 14
11 0 4 13
15 7 3 2
Input Error! (1,3) and (2,3) are inconsistent.
```



Inconsistent: There is only ONE pair of inconsistent cell. Please put the smaller index of the cell in front of the other. You don't need to check the boundary of the maze, since there will always exist doors except the entrance and exit of the 4 edges. (It is OK if you want to check the boundary in case.)

Sample Input and **output 3**:

```
5 6
8 10 10 12 15 15
5 9 12 1 14 15
5 3 6 7 11 14
3 12 13 15 13 15
15 3 6 11 2 10
There is no path.
```



NTHU Online Judge information: (TestCase will be updated soon)

Problem ID: 11781

Problem title: 231001_1/21_FinalProject

Guidelines

1. Mark weightings: Correctness 50% + Readability 50%.
 - A. Correctness: Make sure you understand what the program should do in every case (including special cases).
 - B. Readability: Use comments to explain your logic.
2. You are welcome to discuss with each other, but **DO NOT COPY OTHER PEOPLE'S WORK**. Plagiarism is a serious offense. Not only will you get no points in this assignment, but you may also be reported to the university.
3. Program Style:
 - A. Your program should include a number of functions. Their functionality should be well-defined, easily understandable, and clearly documented as comments within the source code.
 - B. The efficiency of your program should be reasonable. However, don't spend too much time just to speed it up while making the code difficult to read.