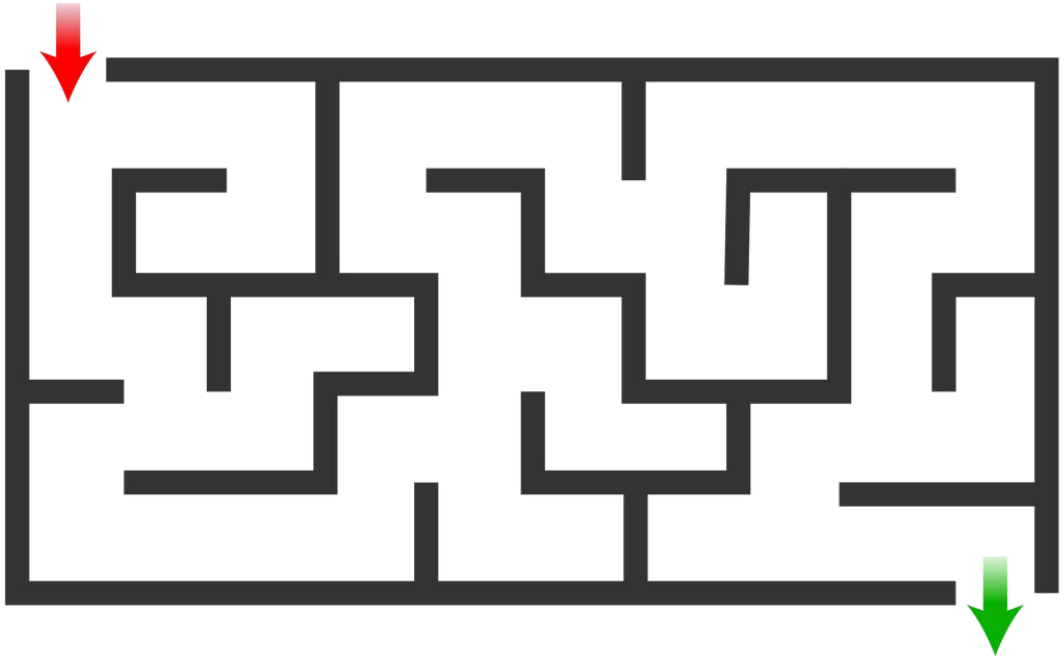
Lab 4

Data Structure : Stack

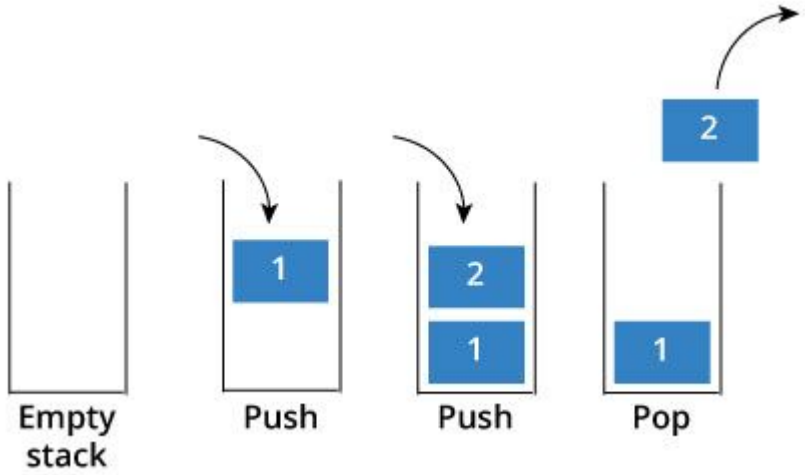
Maze Game - Intro

In this lab, we want you to find out the path from the entrance to the exit in the maze.



Part 1 - Implement Stack by Array

You must use stack when solving this maze game. In this part, you should implement stack by array. Please check the attached file ArrayStack.h and make it working.



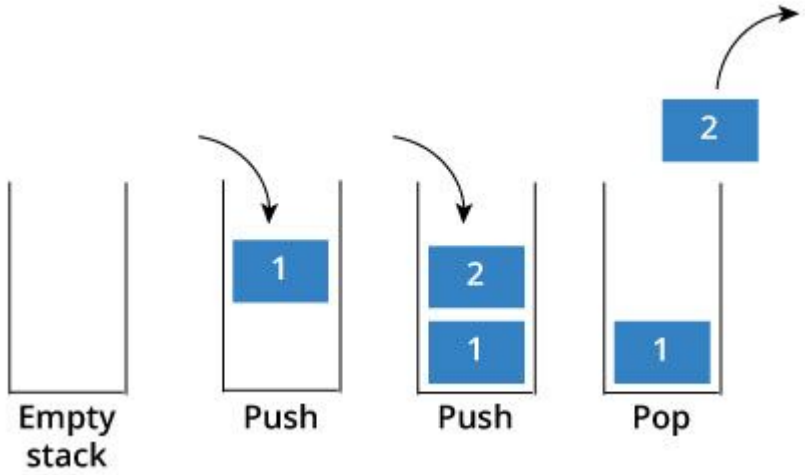
Part 1 - Implement Stack by Array

You should implement 8 functions in the ArrayStack class.

1. constructor (the initial capacity is 1, don’t change it !!!)
2. push : push a node in the stack
3. pop : pop the top node in the stack
4. isEmpty: return whether the stack is empty or not
5. isFull : return whether the stack is full or not
6. getSize : return how many nodes are used in the stack
7. Top : return the data in the top of the stack
8. doubleCapacity : when the array stack is full, double the array stack capacity

Part 2 - Implement Stack by Linked list

You must use stack when solving this maze game. Please check the attached file ListStack.h and complete the code. You should make the stack working.



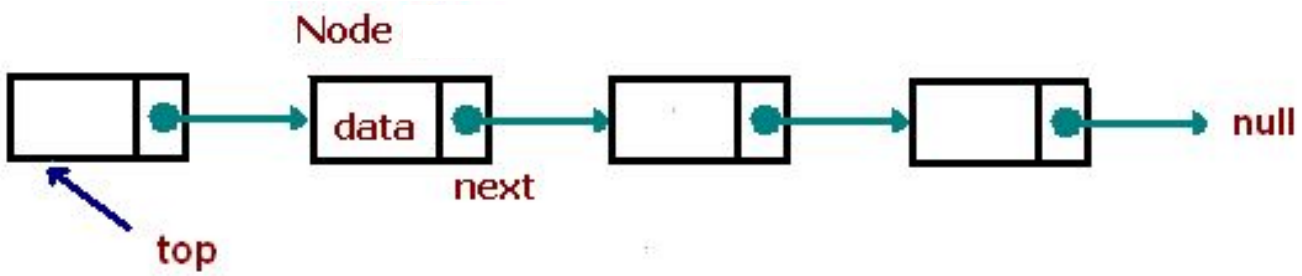
Part 2 - Implement Stack by Linked list

You should implement 6 functions in the ListStack class.

1. constructor
2. push : push a node in the stack
3. pop : pop the top node in the stack
4. isEmpty : return whether the stack is empty or not
5. getSize : return how many nodes are in the stack
6. Top : return the data in the top node

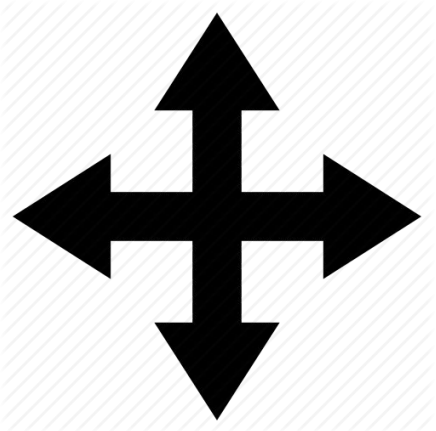
Part 2 - Implement Stack by Linked list

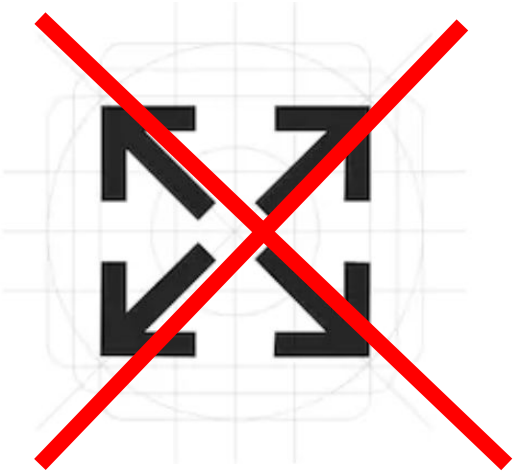
Hint :



Part 3 - Maze Game Rule

You could only walk in four directions : up, down, left, right.





Part 3 - Maze Game Input and Output

We'll give you a MxN maze map(M >= 1, N >= 1), and the position of entrance and exit are in (1,1) and (m,n). Number 0 represent the road and number 1 represent the wall. For example, below is a 5x4 maze,the entrance is as (1,1) and the exit is at (5,4). Please find the shortest path and label the path by *.

Input :

entrance

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

exit

Output :

```
* 0 0 0
* 1 1 0
* * * 1
0 1 * *
0 0 1 *
```

Part 3 - Maze Game No Way

If there’s no way to reach the exit, just output “NOWAY”.

Input :
0 1 0 0
0 0 1 0

Output :
NOWAY

Part 3 - Maze Game Input and Output

In judge system, we'll give you the standard input text on the right, for instance.

In the first line 2 means there are two mazes in this trial.

4,3 means <m,n> maze in the following and 2,5 as well.

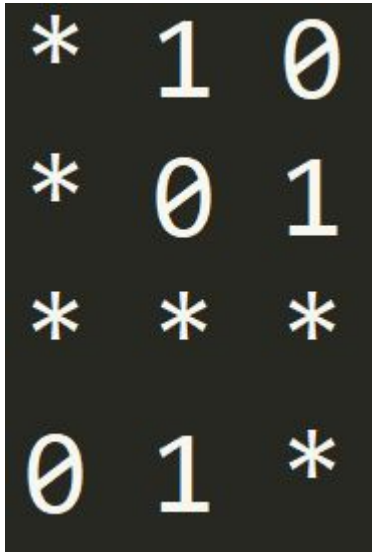
You should output the belowing in standard output.

```
* 1 0
* 0 1
* * *
0 1 *
NOWAY
```

```
2
4 3
0 1 0
0 0 1
0 0 0
0 1 0
2 5
0 1 0 1 0
0 0 0 1 0
```

Part 3 - Maze Game Issue

You may find out two way in this maze. Both are okay because the steps are the same. Just make sure that you find the shortest path. Another issue you may concern is that all of the mazes will only have one path to reach the exit.



Part 3 - Maze Game Hint

Use either ArrayStack or ListStack to Keep Pass History. You must use stack in part 1 or part 2 to finish the maze game.

Other methods such as recursion are forbidden.

You could check the attached file input and output for example.

```
typedef struct {
   int x;
   int y;
   int dir;
} item;
item stack[m*p];
```

Reminder

1. (30 points) In part 1, please upload ArrayStack.h file to e3.
2. (30 points) In part 2, please upload ListStack.h file to e3.
3. (40 points) In part 3, please upload {StudentNumber}.cpp file to e3.
4. Please use linux1~linux4.cs.nctu.edu.tw server to demo.
5. There's only one time for you to demo your code. Don't let TAs down.
6. Feel free to ask any problem, but remember to think before you ask.
7. Don't cheat unless you want to get 0 in this lab.