Assignment 7

This assignment is to be done individually.

Important Note: The university policy on academic dishonesty (cheating) will be taken very seriously in this course. You may not provide or use any solution, in whole or in part, to or by another student.

You are encouraged to discuss the concepts involved in the questions with other students. If you are in doubt as to what constitutes acceptable discussion, please ask! Further, please take advantage of office hours offered by the instructor and the TA if you are having difficulties with this assignment.

DO NOT:

- Give/receive code or proofs to/from other students
- Use search engines to find solutions for the assignment

DO:

- Meet with other students to discuss assignment (it is best not to take any notes during such meetings, and to re-work assignment on your own)
- Use online resources (e.g. Wikipedia) to understand the concepts needed to solve the assignment

Submission Instructions:

- Submit two files on CourSys
  1. `report.pdf`, which contains a write-up of your solutions to the assignment.
  2. `Maze.cpp`, which contains the code you wrote for question 2.

Assignment 7

Question 1 (8 marks)

Convert the following expression to its Postfix format:

$((A + B) - C \times (D/E)) + F$

Explain your steps and stack states.

Question 2 (17 marks)

You are given a maze as $N \times M$ binary matrix of blocks and there is a rabbit initially at $(0, 0)$ and the rabbit wants to eat carrot which is present at some given block (fx, fy) in the maze. In a maze matrix, 0 means that the block is a dead end and 1 means that the block can be used in the path. The rabbit can move in any non-diagonal direction to the blocks that are not dead end.

Your task is to check if there exists any path so that the rabbit can reach the carrot or not. It is not needed to print the path. **You must use the built-in stack class to solve this question.**

Input and output

The program takes several inputs.

1. number of rows (N) and columns (M) of the maze
2. 2D array representing the maze
3. position of the carrot (row, column)

If there is a path from rabbit to carrot the program should print "Path Found", Otherwise it should print "No Path Found".

Starter code

Starter code is provided in `Maze.cpp`. Please complete the `isReachable` function. You may use and modify the provided `main` function for testing.

```
1 #include <cstring>
2 #include <iostream>
3 #include <stack>
4
5 #define N 4
```

```
6  #define M 5
7
8  using namespace std;
9
10 class node {
11 public:
12     int x, y;
13
14     node(int i, int j) {
15         x = i;
16         y = j;
17     }
18 };
19
20
21 bool isReachable(int maze[N][M], int fx, int fy) {
22     stack<node> s; // stack of nodes to go through
23     int i, j; // current location
24
25     // Initially starting at (0, 0).
26     s.push(node(0,0));
27
28     // COMPLETE THIS FUNCTION BELOW
29 }
30
31 // Driver code
32 int main() {
33     // Maze matrix
34     int maze[N][M] = {
35         {1,0,1,1,1},
36         {1,1,1,0,1},
37         {0,0,0,0,1},
38         {1,1,1,0,1}
39     };
40
41     int fx = 3;
42     int fy = 4;
43     //int fy = 2;
44
45     /*
46     int maze[N][M] = {
47         {1,0,1,1,0},
```

```
48          {1,1,1,0,1},
49          {0,1,0,1,1},
50          {1,1,1,1,1}
51      };
52
53      // Food coordinates
54      int fx = 2;
55      int fy = 3;
56      */
57
58      if (isReachable(maze, fx, fy)) {
59          cout << "Path Found!" << '\n';
60      } else {
61          cout << "No Path Found!" << '\n';
62      }
63
64      return 0;
65  }
```

Example 1:

Input:

```
1  4,5
2  1,0,1,1,1
3  1,1,1,0,1
4  0,0,0,0,1
5  1,1,1,0,1
6  3,4
```

Output: `Path Found`

Example 2:

Input:

```
1  4,5
2  1,0,1,1,1
3  1,1,1,0,1
4  0,0,0,0,1
5  1,1,1,0,1
6  3,2
```

Output: `No Path Found`