Lab Test 2

Algorithm Analysis and Design Lab

Course Code: ICT-2202

Problem 1:

Given an integer **N** representing the number of pairs of parentheses, the task is to generate all combinations of well-formed(balanced) parentheses.

Example 1:

```
Input:
N = 3
Output:
((()))
(()())
(())()
()(())
()()()
```

Example 2:

```
Input:
N = 1
Output:
()
```

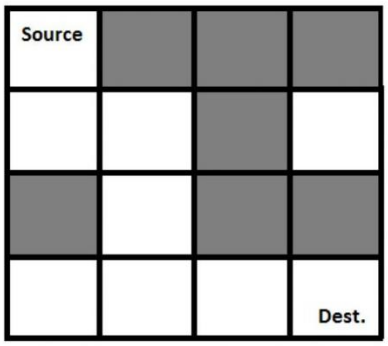
Problem 2:

A Maze is given as N*N binary matrix of blocks where source block is the upper left most block i.e., maze[0][0] and destination block is lower rightmost block i.e., maze[N-1][N-1]. A rat starts from source and has to reach the destination. The rat can move only in two directions: forward and down.

In the maze matrix, 0 means the block is a dead end and 1 means the block can be used in the path from source to destination. Note that this is a simple version of the typical Maze problem.

For example, a more complex version can be that the rat can move in 4 directions and a more complex version can be with a limited number of moves.

Following is an example maze.
Gray blocks are dead ends (value = 0).



Following is a binary matrix representation of the above maze.

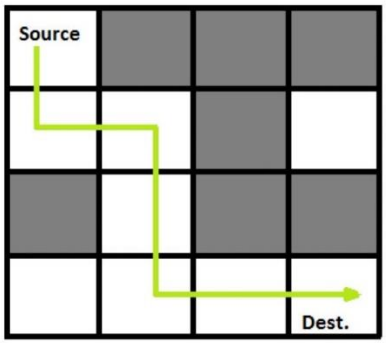
{1, 0, 0, 0}

{1, 1, 0, 1}

{0, 1, 0, 0}

{1, 1, 1, 1}

Following is a maze with highlighted solution path.



Following is the solution matrix (output of program) for the above input matrix.

{1, 0, 0, 0}

{1, 1, 0, 0}

{0, 1, 0, 0}

{0, 1, 1, 1}

All entries in solution path are marked as 1.