VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

VIT Vidyalankar Institute of Technology
Accredited A+ by NAAC

DEPARTMENT OF COMPUTER ENGINEERING

Experiment No. 8

Semester	T.E. Semester VI
Subject	ARTIFICIAL INTELLIGENCE (CSL 604)
Subject Professor In-charge	Prof. Avinash Shrivas
Assisting Teachers	Prof. Avinash Shrivas

Student Name	Deep Salunkhe
Roll Number	21102A0014
Lab Number	310A

Title:

Bot saves princess

Theory:

Princess Peach is trapped in one of the four corners of a square grid. You are in the center of the grid and can move one step at a time in any of the four directions. Can you rescue the princess?

Input format

The first line contains an odd integer N (3 <= N < 100) denoting the size of the grid. This is followed by an NxN grid. Each cell is denoted by '-' (ascii value: 45). The bot position is denoted by 'm' and the princess position is denoted by 'p'.

Grid is indexed using Matrix Convention

Output format

Print out the moves you will take to rescue the princess in one go. The moves must be separated by '\n', a newline. The valid moves are LEFT or RIGHT or UP or DOWN.

Sample input

```
3
---
-m-
p--
```

Sample output

```
DOWN
LEFT
```

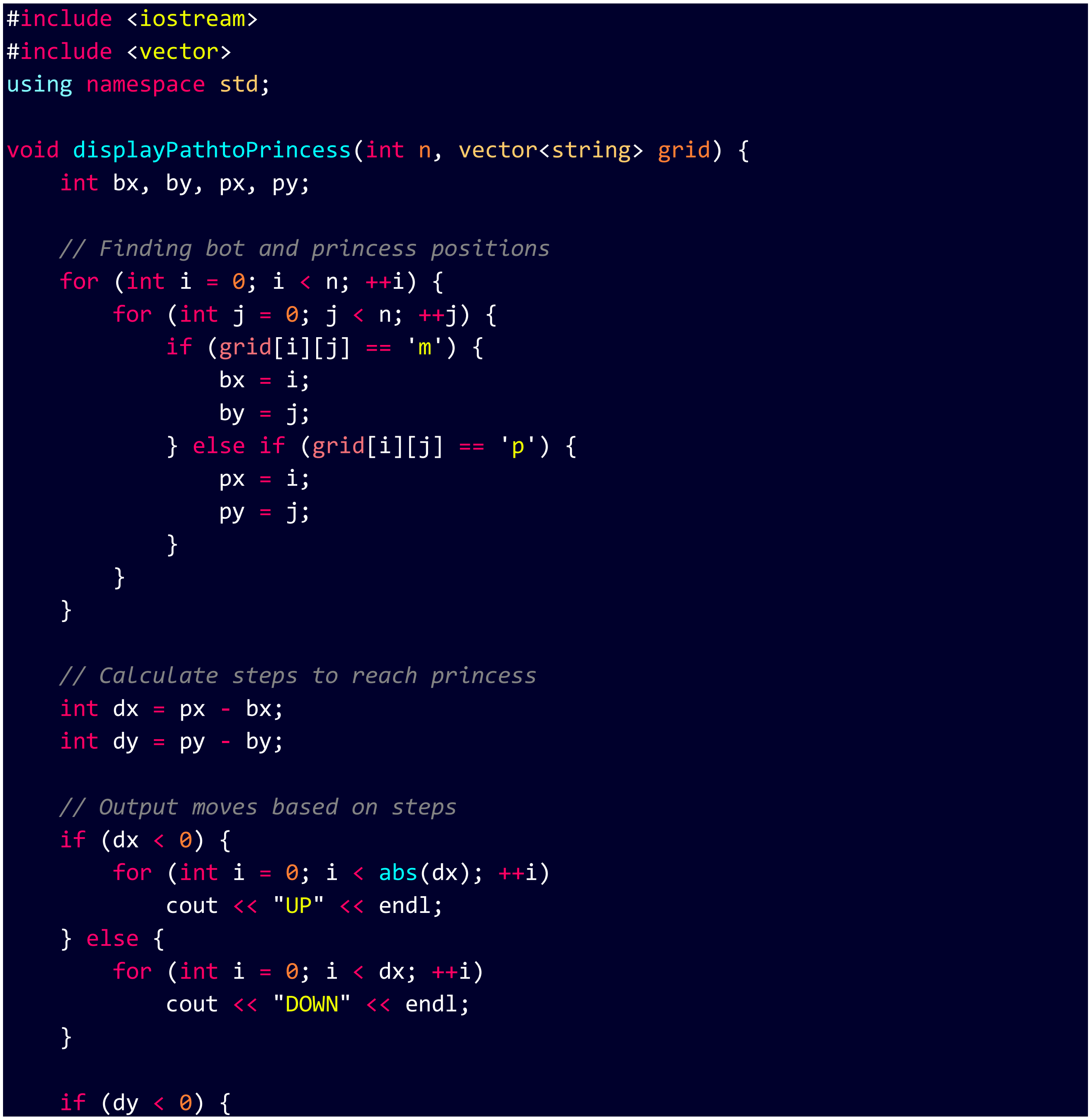
Task

Complete the function *displayPathtoPrincess* which takes in two parameters - the integer N and the character array grid. The grid will be formatted exactly as you see it in the input, so for the sample input the princess is at grid[2][0]. The function shall output moves (LEFT, RIGHT, UP or DOWN) on consecutive lines to rescue/reach the princess. The goal is to reach the princess in as few moves as possible.
The above sample input is just to help you understand the format. The princess ('p') can be in any one of the four corners.

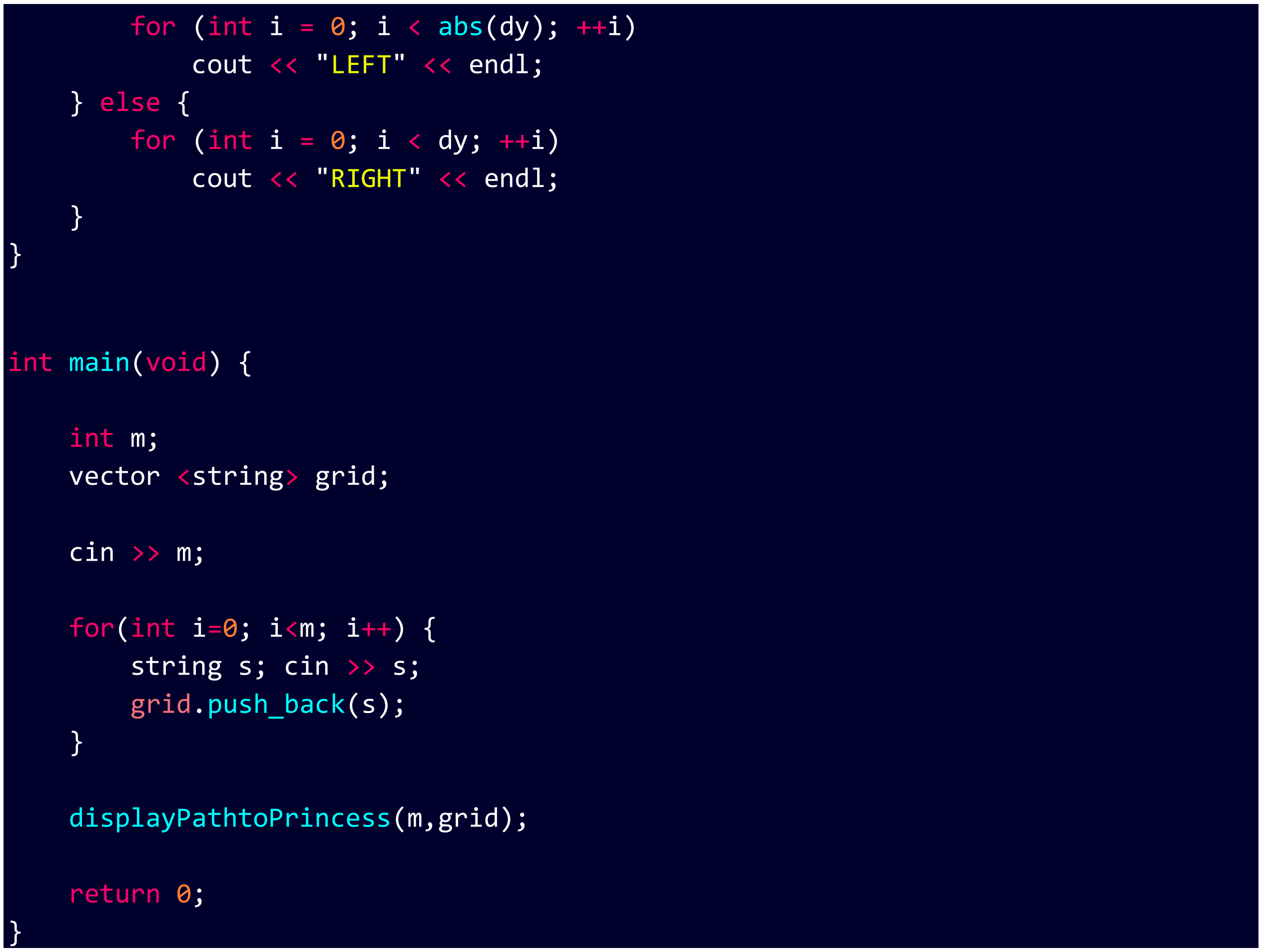
Scoring

Your score is calculated as follows : (NxN - number of moves made to rescue the princess)/10, where N is the size of the grid (3x3 in the sample testcase).

Program Code:

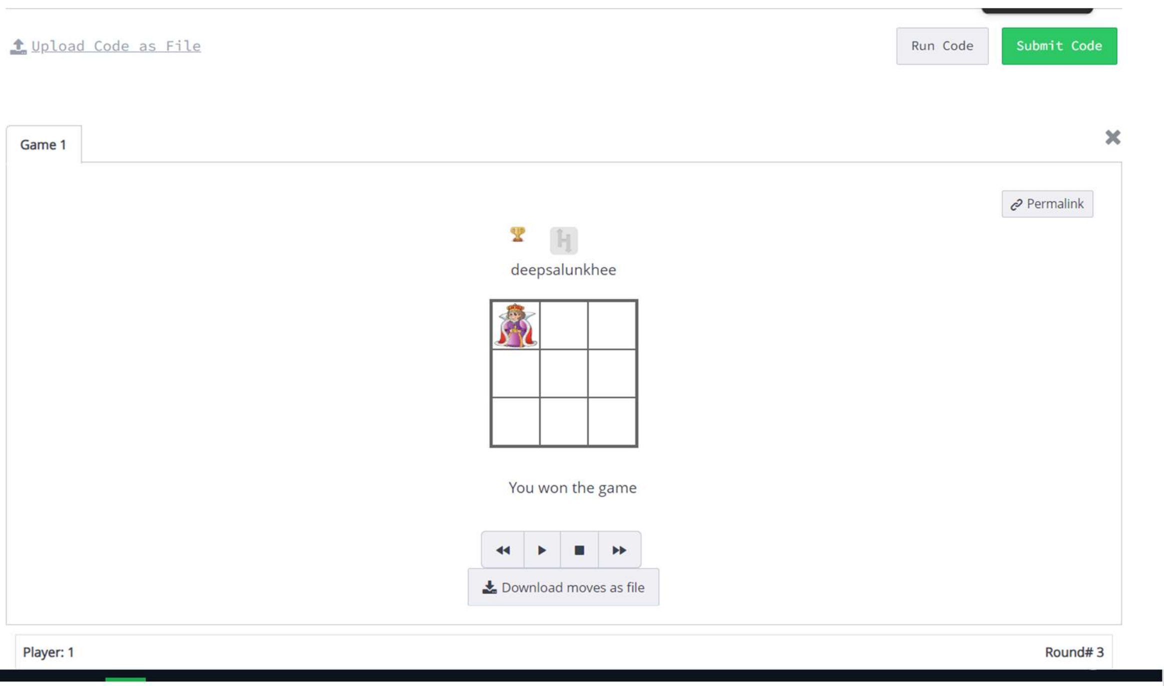


```
#include <iostream>
#include <vector>
using namespace std;

void displayPathtoPrincess(int n, vector<string> grid) {
    int bx, by, px, py;

    // Finding bot and princess positions
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            if (grid[i][j] == 'm') {
                bx = i;
                by = j;
            } else if (grid[i][j] == 'p') {
                px = i;
                py = j;
            }
        }
    }

    // Calculate steps to reach princess
    int dx = px - bx;
    int dy = py - by;

    // Output moves based on steps
    if (dx < 0) {
        for (int i = 0; i < abs(dx); ++i)
            cout << "UP" << endl;
    } else {
        for (int i = 0; i < dx; ++i)
            cout << "DOWN" << endl;
    }

    if (dy < 0) {
```



```
        for (int i = 0; i < abs(dy); ++i)
            cout << "LEFT" << endl;
    } else {
        for (int i = 0; i < dy; ++i)
            cout << "RIGHT" << endl;
    }
}


int main(void) {

    int m;
    vector <string> grid;

    cin >> m;

    for(int i=0; i<m; i++) {
        string s; cin >> s;
        grid.push_back(s);
    }

    displayPathtoPrincess(m,grid);

    return 0;
}
```

Output:



Upload Code as File

Run Code

Submit Code

Game 1

Permalink

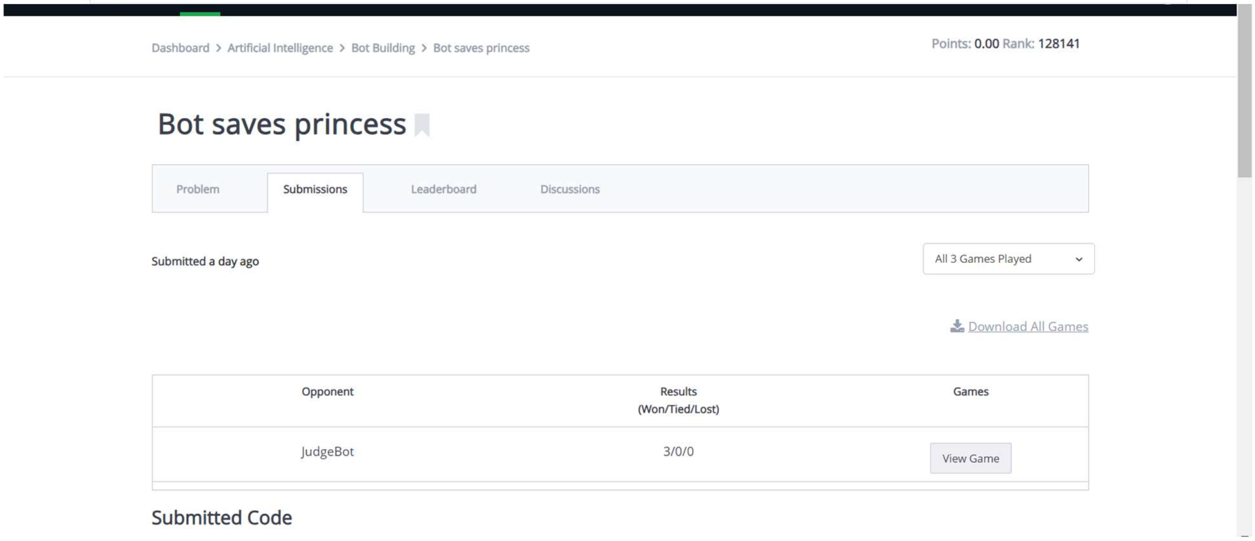
deepsalunkhee

You won the game

Download moves as file

Player: 1

Round# 3



Dashboard > Artificial Intelligence > Bot Building > Bot saves princess

Points: 0.00 Rank: 128141

Bot saves princess

Problem | Submissions | Leaderboard | Discussions

Submitted a day ago

All 3 Games Played

Download All Games

Opponent	Results (Won/Tied/Lost)	Games
JudgeBot	3/0/0	View Game

Submitted Code