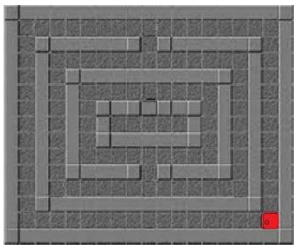
CS 218 DATA STRUCTURES
ASSIGNMENT 2
SECTION C and F
Fall 2022

DUE: Thursday, 17 Oct, 2022 (*Late submissions will not be accepted)*

SUBMIT documented and well-written structured code in C++ in the classroom. Undocumented code will be assigned a zero.

Problem

We wish to develop a game, " Treasure Hunt in a Haunted House." The user is outside a haunted house to look for the treasure, which is hidden in one of the rooms. The haunted house has many spooky rooms connected by Portals that can teleport users randomly from one room to another. The rooms are guarded by devils, but they also have a few friendly ghosts, Gaspers flying around. If you come in contact with a Gasper, it can teleport you to the room with treasure. But if you encounter a devil, it would eat you up alive.

The rooms are in the form of a maze, so we represent each room in the haunted house as a 2D matrix, where the first row and column represent the coordinates.

The alphabets and numbers in the matrix have the following meanings:
X indicates => a forbidden cell (Wall),
0 indicates => an allowed cell (user can move to this cell),
D=> Devil, G=> Gasper, T=> Treasure,
P= > Portal teleports the user randomly to another room to a random cell.
Every time user reaches a portal. The portal flips a coin. If the head comes, he is transported randomly to some room in the house. In case the tail occurs then he is taken back to the room from where he came (to the cell next to portal).
Note: For the first time, when the user enters the house and reaches a portal in a room, then on the tail, he is kicked out of the house, and the game ends.

	0	**1**	**2**	**3**	**4**	**5**	**6**
0	X	X	0	0	0	0	0
1	X	X	**D**	X	0	X	X
2	X	X	0	X	0	X	X
3	X	X	X	X	0	X	X
4	0	0	0	0	0	0	X
5	0	X	X	X	X	0	X
6	0	X	X	X	X	0	0
7	0	0	X	X	X	0	G
8	X	0	0	0	X	0	0
9	X	0	0	**T**	X	**P**	0
10	X	X	X	X	X	X	X

Example of a room map in a haunted house

The user has to find a path from a starting point to the treasure going through the allowed cells. For example, in the above, if (0,2) is the start state and (9,3) is the goal state (Treasure), then one possible path is given by the blue highlighted cells.

A user can move {up,down,left,right} from current position. If a user encounters a devil in the path, it dies, and the game ends. If it encounters a Gasper, then Gasper teleports the user to a room with treasure, and the user starts the search there. And if a user reaches a portal, then he is teleported to another room or taken to the previous one, as described above.

FORMAT OF the INPUT FILE

First, set up the haunted house using the file hauntedhouse.txt. The file contains an integer for the number of rooms, followed by the name of the files that contains each room map. In the room file, the first line contains the room dimension, i.e. the number of rows and columns and the second line contains the starting cell coordinates. The characters are separated by space in the file.

Example Input:

hauntedhouse.txt

```
2
Room1.txt
Room2.txt
```

Room1.txt

```
9, 6
0, 2
X X 0 0 0 0
X X D X 0 X
X X 0 X 0 X
X X X X 0 X
0 0 0 0 0 0
0 X X X X 0
0 X X X X P
0 0 X X X 0
X 0 0 G X 0
```

Room2.txt

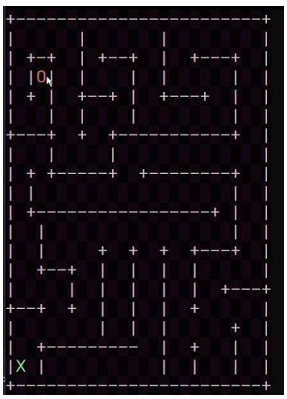
```
7, 5
1, 3
X X X 0 0
X X 0 0 0
X X 0 X 0
X X 0 X 0
0 0 0 P 0
0 X X X X
T X X X X
```

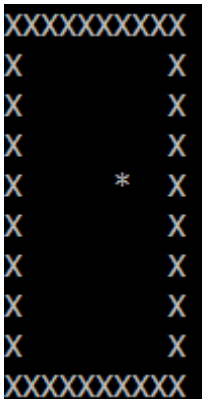
Load the map of each room from the room file into a 2D matrix. Start the user randomly from any room (given starting cell). Display only the room in which the user currently exists on the screen. The user can move {Left, Right, Up, and Down} using arrow keys. The game ends when users find the treasure (T), or is eaten up by a devil, or hit the Q (quit) key.

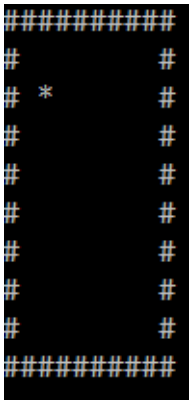
YOU HAVE TO:

- Provide the **iterative implementation** of the above game using the stack template.
- Provide the **recursive implementation** of the above game.
- **If the treasure is found**,
  - print the coordinates of the cells along with the room number where the treasure exists.
  - Also, print the path from starting cell to the treasure. Printing paths means printing coordinates of the cells from starting until you reach the cell with the treasure. Hint use stack.

The format of the input file should be as given above. However, you can be creative with your display on the screen.







Consider the following code if you are not familiar with moving objects on the screen with up, down, left right arrow keys

```
#include<iostream>
```

```
#include <conio.h>
using namespace std;
#define KEY_UP 72
#define KEY_DOWN 80
#define KEY_LEFT 75
#define KEY_RIGHT 77
#define N  10
#define M  10

void Print(int r, int c) {
	system("cls");
	char X = '#', S = ' ';
	char map[N][M] = { {X, X, X, X ,X, X, X, X, X ,X},
			   {X, S, S, S ,S, S, S, S, S ,X},
			   {X, S, S, S ,S, S, S, S, S ,X},
			   {X, S, S, S ,S, S, S, S, S ,X},
			   {X, S, S, S ,S, S, S, S, S ,X},
			   {X, S, S, S ,S, S, S, S, S ,X},
			   {X, S, S, S ,S, S, S, S, S ,X},
			   {X, S, S, S ,S, S, S, S, S ,X},
			   {X, S, S, S ,S, S, S, S, S ,X},
			   {X, X, X, X ,X, X, X, X, X ,X} };

      for (int i = 0; i < N; i++) {
		for (int j = 0; j < M; j++)
			if (i == r && j == c) cout << "*";
			else cout << map[i][j];
		cout << endl;
	}
}

int main(){
	int c = 0;
	int x=2, y=2;
	Print(x, y);
	while (1)
	{
		c = 0;
		switch ((c = _getch())) {
		case KEY_UP:
			if(x>1) x = x - 1;
			Print(x,y);
			break;
		case KEY_DOWN:
			if (x < N-2) x = x + 1;
			Print(x, y);
			break;
		case KEY_LEFT:
			if (y > 1) y = y - 1;
			Print(x, y);
			break;
		case KEY_RIGHT:
			if (y < M-2) y = y + 1;
			Print(x, y);
			break;
		default:
			break;
		}

	}
	system("Pause");
	return 0;
}
```