**CSE102** **HW07**

## Part 1 100pts

While our spy, Ali, runs away from the enemy base after a failed infiltration operation, he is shot by the enemy commander. Fortunately, he does not die but passes out. After a while, Ali opens his eyes and he figures out that he is in a maze. The enemy commander gives him a chance to live in one condition – He must exit from the maze to live. Because Ali gives no damage to the enemy. There is one thing that the commander does not know. You, the tech guy, hacked the CCTV system of the maze and you will help Ali to escape from the maze.

The maze built from tiles that has three different attributes. These are explained below.

- Distance: Tiles has different road length. The **distance** value is integer.
- Comfort Level: Some of the tiles are made with stone and some other tiles are made with sand. So, Ali can walk on the stone tiles easier than sand. The **comfortLevel** value is integer.
- Danger Level: Some of the tiles are more dangerous from other tiles. The **dangerLevel** value is integer.

In addition to these attributes, the commander sets up some traps on the maze. If there is a trap on a tile and you pass through that tile, you will take damage of the value that trap can give. The **trapDamage** value is integer.

You will calculate and three types of escape route and give them to Ali. These are described below.

- Shortest Path: The route that minimize the total distance value.
- Comfortable Path: The route that minimize the total comfortLevel value.
- Safest Path: The route that minimize the total dangerLevel value.

Beware that Ali is wounded and bleeding. He has a MAX_LIFE and the it decreases over time. The time step in this problem is a tile. If you move by one tile Ali's life decreases by one unit.

- Shortest Path: Decreases **4 unit**.
- Comfortable Path: Decreases **2 unit**.
- Safest Path: Decreases **6 unit**.

**If Ali caught by a trap, Ali's life decreases by the amount of traps damage.**

You can only use recursion to calculate routes. There is no loop for these functions.

There will be a base code to initialize and print maze and other stuff.

**STARTER CODE: https://goo.gl/4UsLhp**

### Signature

```
typedef enum _tileType {BORDER, WALL, EMPTY, USED, START} TILE_TYPE;
typedef enum _pathType {DISTANCE, COMFORT, DANGER} PATH_TYPE;
typedef struct _tiles{
	TILE_TYPE type;
	int distance;
	int dangerLevel;
	int comfortLevel;
	int trapDamage;
}TILES;
typedef struct _coord{
	int x;
	int y;
}Coord;
```

```
typedef struct _path{
	Coord coords[200];
	int size;
	int totalDistance;
	int totalDanger;
	int totalComfort;
	int totalDamage;
}Path;

int isStuck(TILES maze[100][100], Coord currentTile);
int isExit(TILES maze[100][100], Coord currentTile);

Path shortestExit(TILES maze[100][100], Coord currentTile, Path path, int *minWeight);
Path comfortableExit(TILES maze[100][100], Coord currentTile, Path path, int
*minWeight);
Path safestExit(TILES maze[100][100], Coord currentTile, Path path, int *minWeight);
```

## Sample Usage

In the starter code

## Return Value

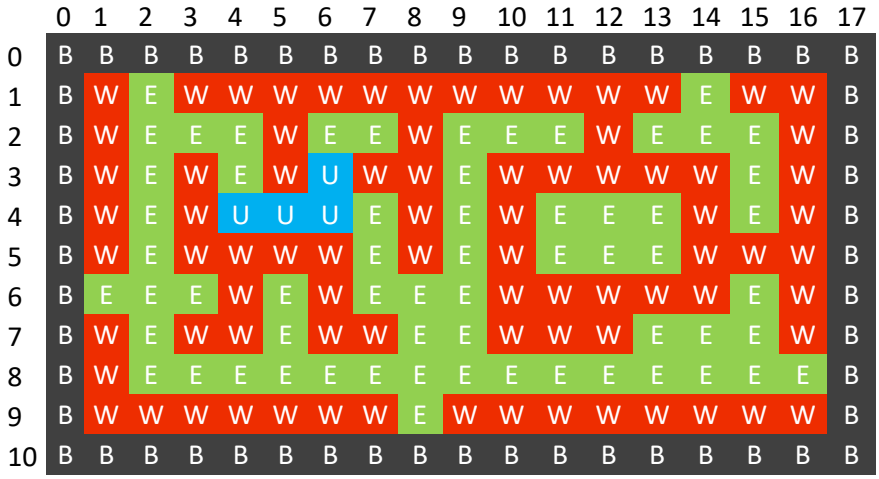


*Figure 1: Sample maze*

```
Board name: maze1.txt
Shortest Path
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
0 B B B B B B B B B B B B B B B B B B
1 B   3                       2     B
2 B   2 7 6   2 2   6 7 3   2 2 2   B
3 B   4   1   2     5           2   B
4 B   7   1 1 1 1   1   4 4 4   2   B
5 B   4         1   1   4 4 4       B
6 B 9 1 1   1   1 1 1           5   B
7 B   2     1     9 1       3 4 2   B
8 B   3 5 9 2 6 9 1 1 1 1 1 1 1 1 1 B
9 B               1                 B
0 B B B B B B B B B B B B B B B B B B

  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
0 B B B B B B B B B B B B B B B B B B
1 B W . W W W W W W W W W W   W W B
2 B W .     W     W       W       W B
3 B W . W   W   W W   W W W W W   W B
4 B W . W         W   W       W   W B
5 B W * W W W W   W   W       W W W B
6 B       W   W       W W W W W   W B
7 B W   W W   W W     W W W       W B
8 B W                             B
9 B W W W W W W W   W W W W W W W W B
0 B B B B B B B B B B B B B B B B B B

Total weight: 20
-------------------------------------
Board name: maze1.txt
Comfortable Path
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
0 B B B B B B B B B B B B B B B B B B
1 B   3                       2     B
2 B   1 7 6   2 2   3 6 3   1 4 3   B
3 B   2   1   2     5           2   B
4 B   3   1 1 1 1   1   4 4 4   6   B
5 B   4         1   1   4 4 4       B
6 B 1 6 4   1   1 1 1           5   B
7 B   1     1     9 1       3 4 2   B
8 B   1 3 2 7 0 1 1 8 8 8 8 8 8 8 8 B
9 B               1                 B
0 B B B B B B B B B B B B B B B B B B
```

```
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
0 B B B B B B B B B B B B B B B B B B
1 B W . W W W W W W W W W W   W W B
2 B W .     W     W       W       W B
3 B W . W   W   W W   W W W W W   W B
4 B W . W         W   W       W   W B
5 B W * W W W W   W   W       W W W B
6 B       W   W       W W W W W   W B
7 B W   W W   W W     W W W       W B
8 B W                             B
9 B W W W W W W W   W W W W W W W W B
0 B B B B B B B B B B B B B B B B B B

Total weight: 13
-------------------------------------
Board name: maze1.txt
Safest Path
  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
0 B B B B B B B B B B B B B B B B B B
1 B   1                       2     B
2 B   3 1 1   2 2   6 7 3   2 2 2   B
3 B   5   1   2     5           2   B
4 B   7   1 1 1 1   1   4 4 4   2   B
5 B   9         1   1   4 4 4       B
6 B 9 9 1   1   1 1 1           5   B
7 B   2     1     6 8       3 4 2   B
8 B   3 1 1 1 2 3 2 7 1 1 4 5 4 1 1 B
9 B               3                 B
0 B B B B B B B B B B B B B B B B B B

  0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7
0 B B B B B B B B B B B B B B B B B B
1 B W . W W W W W W W W W W   W W B
2 B W .     W     W       W       W B
3 B W . W   W   W W   W W W W W   W B
4 B W . W         W   W       W   W B
5 B W * W W W W   W   W       W W W B
6 B       W   W       W W W W W   W B
7 B W   W W   W W     W W W       W B
8 B W                             B
9 B W W W W W W W   W W W W W W W W B
0 B B B B B B B B B B B B B B B B B B

Total weight: 25
```