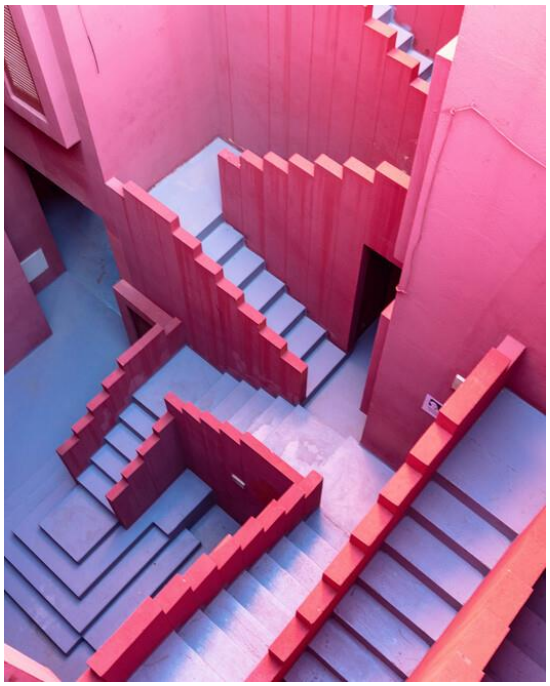
# Lab 10



The goal of the lab is to escape from the maze. There are 5 mazes provided that you can test your algorithm on.

You are allowed 200 moves in the maze before you die. You will appear at some point in the maze but you don't know where. Each maze has exactly one exit.

Each maze is a 20x20 grid consisting of walls and spaces. The main method that reads in the data is provided (see Lab10 code below). The input reads in the maze structure from a series of 20 Strings, followed by two int co-ordinates, giving your X and Y positions in the 20x20 maze.

All you need to do is complete the getMove() method, which takes in 4 booleans representing if it's possible to move north, south, east and west (true means you can, false means there's a wall).

The program will print out the maze every time you make a step and then pause for 100ms. If you move into a wall, nothing happens but

you lose a life. The program will terminate if you find the exit or lose all your lives.

```
import java.util.*;
import java.io.*;

public class Lab10{

    public static void main (String[] args){
        File file = new File("C:\\ Maze1.txt");

        int lives = 200;
        int posX = 0;
        int posY = 0;
        String[] input = new String[20];
        try {
            Scanner scan = new Scanner(file);

            for(int i = 0; i < 20; i++) {
                input[i] = scan.nextLine(); // read the maze
            }
            posX=Integer.parseInt(scan.nextLine()); //this is where
you appear
            posY=Integer.parseInt(scan.nextLine());
            scan.close();
        } catch (Exception e) {
            System.err.println(e);
        }

        boolean[][] maze = new boolean[20][20];
        for(int i=0;i<20;i++){
            for(int j=0;j<20;j++){
```

```
                if(input[i].charAt(j)=='X'){
                    maze[i][j]=false; //there's a wall
                }else{
                    maze[i][j]=true; //there's a space
                }
            }
        }
        Brain myBrain = new Brain();


       while(lives>0){
            System.out.println("Current position: "+posX+" "+posY);
            for(int i=0;i<20;i++){ //print out the map
                for(int j=0;j<20;j++){
                    if(posX==i&&posY==j){
                        System.out.print("o");
                    }else if(maze[i][j]==true){
                        System.out.print(" "); //there is a space
                    }else{
                        System.out.print("X"); //there is a wall
                    }
                }
                System.out.println();
            }

            try{
                Thread.sleep(100);
            }catch (InterruptedException e) {
                // Handle interrupted exception (if necessary)
                e.printStackTrace();
            }
            String move =myBrain.getMove(maze[posX-
1][posY],maze[posX+1][posY],maze[posX][posY+1],maze[posX][posY-1]);
```

```
            if(move.equals("north")&&maze[posX-1][posY]){
                posX--;  //if the brain wants to move North AND it's
possible
            }else if(move.equals("south")&&maze[posX+1][posY]){
                posX++;  //if the brain wants to move South AND it's
possible
            }else if(move.equals("east")&&maze[posX][posY+1]){
                posY++;
            }else if(move.equals("west")&&maze[posX][posY-1]){
                posY--;
            }
            lives--;
            if(posY%19==0||posX%19==0){ //found a way out!
                System.out.println("You found the exit at:
"+posX+","+posY);
                System.exit(0);
            }
        }
        System.out.println("You died in the maze!");
    }
}


class Brain{
    //this is the dumbest possible strategy - random
    public String getMove(boolean north, boolean south, boolean
east, boolean west){
        int random = (int)(Math.random()*4);
        switch(random){
            case 0: return "north";
            case 1: return "south";
            case 2: return "east";
            default: return "west";
        }
    }
```

}