Q. 1)

Consider a grid map of size m×n. The origin (0, 0) is taken as the top-left corner. The X axis is the vertical axis, the Y axis is the horizontal axis. A cell 0 denotes obstacle-free region, -1 denotes obstacles, and 1 denotes a gem. Given a source, the aim of the agent is to go to the gem. The step cost is the time. The agent moves 1 block distance per second (or $\sqrt{2}$ time for diagonal moves). The order of preference of actions is: E, SE, S, SW, W, NW, N, NE, where N, E, S, W stand for North, East, South and West respectively. The priority queue should be implemented such that, in case of a tie, a node that is inserted first should be removed first. Print the path from the source to the gem. Print only -1 if no path exists.
The input is t, the number of test cases, followed by (for every test case) m and n. The next m lines of n integers each denote the grid map. The next line contains the source indicated by two integers Sx Sy. The output is the path.

Q. 2)

Solve (1) with the difference that the map is circular in both X and Y axis, and moving right of the last column places the robot on the 1st column, and similarly with the rows.

Q. 3)

Solve (1) with the constraint that the robot can only move in the direction in which it is facing. If the robot wants to change its direction, it must invoke the action turn, which turns the robot by 45 degrees clockwise or anticlockwise. The turning action has a cost of 5. The robot is initially facing positive the Y axis (looking E).

Q. 4)

Consider a *n*×*n* size board (1≤n≤10), wherein every cell has a number written on it. The rows and columns are cyclic. So you can slide any row/column any number of blocks or any number of times. As an example in the board given below, we slide the middle row one block each repeatedly, and the results are given in the following figures.

A unit move in this game is sliding any 1 row (or 1 one column) any number of times. So the motions made above is one single move of the game. Given a source configuration and a goal configuration, print the moves that result in the goal configuration from a source configuration using Iterative Deepening. Moving rows is preferred to moving columns (primary criterion), an earlier row/column is preferred to a later one (secondary criterion), sliding lesser rightward/downward is preferred to sliding more (tertiary criterion).

4	8	5
2	6	13
1	10	18

4	8	5
13	2	6
1	10	18

4	8	5
6	13	2
1	10	18

4	8	5
2	6	13
1	10	18

The first line of input is the number of test cases. For every test case, the first line of input is n, the size of the game. The next n lines with n integers each is the source configuration. The next n lines with n integers each is the goal configuration.
The output is variable number of lines, each line indicating the board configuration of a move printed in a row major format.

Q. 5)

Consider a grid map of size *m*×*n*. The origin (0, 0) is taken as the top-left corner. The X axis is the vertical axis, the Y axis is the horizontal axis. A cell 0 denotes obstacle-free region, -1 denotes obstacles, and 1 denotes a gem. Given a source, the aim of the agent is to go to the gem. The step cost is the time. The agent moves 1 block distance per second (or √2 time for diagonal moves). The order of preference of actions is: E, SE, S, SW, W, NW, N, NE, where N, E, S, W stand for North, East, South and West respectively. Print the path from the source to the gem using Iterative Lengthening. Print only -1 if no path exists.
The input is t, the number of test cases, followed by (for every test case) m and n. The next m lines of n integers each denote the grid map. The next line contains the source indicated by two integers Sx Sy. The output is the path.
Take ε as 1. 1≤m,n≤100