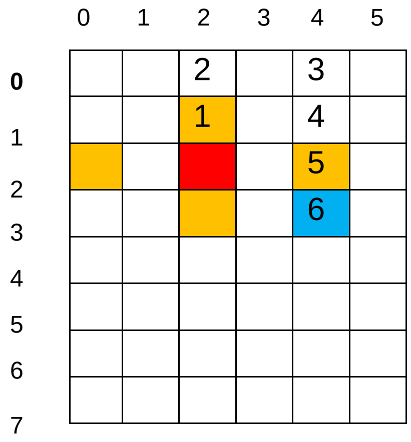
AI/DFS Problem:

Given the below Figure. We have an 8x6 board. The rows and columns are numbered starting from 0. The figure below shows the way to calculate the next move of a new token. The red square at (2, 2) is the current position of the token, and the orange squares are the possible positions resulting from one of the possible moves from red square. You are assumed to use DFS strategy on this board to reach the final state which in the example it is considered as the blue square (3, 4). Please note that we are making an assumption that in an order you will visit top, right, bottom and left. As in the example you can go 1 units to top and bottom, and 2 units to right and left. The numbered squares are an example to the solution for the provided example.



Input: You are required to have a code to accept **problem.txt** file which the first line contains an integer **n** specifying the number of problems in the file. Each line will contain a problem. In each problem line, the first pair of integers is the source position of the token and the next pair specifies the destination.

Please see the following sample **problem.txt** file

Line number	***Contents:***
1.	2
2.	2 2 3 4
3.	2 1 6 3

For example in the sample problem, 2 shows that there are 2 problems, (2, 2) is the start point of the first problem and (3, 4) is the end point of the problem.

Output: All outputs should be written in a file called **solution.txt** in the current directory. For each problem in **problem.txt**, there will be a **corresponding line** in **solution.txt** containing one integer: the minimum number of moves required for the corresponding problem and the solution path and the step number. If there is no solution, please write no solution. The sample **solution.txt** for the sample **problem.txt** is as follows:

Line number	***Contents:***
1.	(2,2)(1,2)(0,2)(0,4)(1,4)(2,4)(3.4) ---6 steps
2.	(2,1)(1,1)(0,1)(0,3)(0,5)(1,5)(2,5)(3,5)(4,5)(5,5)(6,5)(7,5)(7,3)(6,3) ---13 steps