Q1-Q2

Question 1

Max. score: 5

This problem is no longer available for practice. Apology for any inconvenience!

There is a *m*-lane road that is *n* grid wide. A fast self-driving car intends to go through this road as quickly as possible. Assume that the road has *k* vehicles already onto it, however, somehow these vehicles are static (do not move). The self-driving car can either move forward at the rate of 2 grids per unit time, provided no collision occurs. This action has a cost of 1. It may as well turn left or right; in which case it moves 1 grid left/right and 1 grid ahead with a cost of 2. Note that a car can never move purely sideways as per Physics. The vehicle can also move forward at a rate of 1 grid per unit time, an action that has a cost of 1. Calculate the plan to reach the end of the road while minimizing cost.

Notes on feasibility:

- If vehicle being planned intends to go 1 grid ahead, no other vehicle's position at the next time step should be the same as the position of the vehicle being planned at the next time step (1 grid ahead).
- If vehicle being planned intends to go 2 grids ahead, no other vehicle's position at the next time step should be the same as the position of the vehicle being planned after 0.5 step (1 grid ahead) or next step (2 grids ahead).
- If vehicle being planned intends to go left (1 grid forward and 1 left), no other vehicle's position at the next time step should be the same as the position of the vehicle being planned at the next step (1 grid ahead and 1 left). However, since the vehicle will partly occupy the grids (1 ahead 0 left, 1 left 0 ahead) if any other vehicle occupies these grids in the next step, then also the solution is invalid.
- The case for right is the same with the left replaced with right.

Constraints: m (lanes) <=10, n (road length) <=500, k (no of vehicles) <=50

Input Format

The first line contains T, the number of test cases. For each test case, the first line contains (m, the number of lanes), *n* (the length of the road), and *k* (the number of vehicles on the road). The next *k* lines contain the position of the *k* vehicles as lane number (0-indexed) and the grid number along the road length (0-indexed). The next line contains the initial lane number for the self-driving car to plan, while it is always at the 0^{th} grid along the road length.

Output Format

For every test case, print the cost of navigation of the car. In case no solution is possible, print NIL.

Sample Input

```
1
3 20 3
0 8
1 5
2 13
1
```

Sample Output

```
13
```

Sample Path

1 0
0 1
0 2
0 4
0 6
1 7
1 9
1 11
1 13
1 15
1 17
1 19

SAMPLE INPUT	SAMPLE OUTPUT
1 3 20 3 0 8 1 5 2 13 1	13

Time Limit:	5.0 sec(s) for each input file.
Memory Limit:	5560 MB
Source Limit:	1024 KB
Marking Scheme:	Score is assigned if any testcase passes.
Allowed Languages:	C++, C++14, C++17, Java, Java 8, Java 14, Python, Python 3, Python 3.8

Question 2

Max. score: 5

This problem is no longer available for practice. Apology for any inconvenience!

In question 1 print the grid number at which the self-driving car should be after every time step. In case of a tie, a lexicographically smaller path is better (as integers 2<11). In case no solution is possible, print NIL.

Sample Input

1
3 20 3
0 8
1 5
2 13
1

Sample Output

1 0
0 1
0 2
0 4
0 6
1 7
1 9
1 11
1 13
1 15
1 17
1 19

Q3-Q4

Question 3

Max. score: 5

This problem is no longer available for practice. Apology for any inconvenience!

In question 1 assume that the road has k vehicles already onto it that move at a speed of 1 grid per unit time. These vehicles neither turn left, nor turn right and keep moving straight.

Note on feasibility:

Calculate the updated position of all vehicles.

For 1 forward, the next position should be collission free wrt the next position of other vehicles.

For 2 forward, the next 2 positions should be collission-free wrt to the next position of the other vehicles

For 1 forward+left wrt the new positions of all vehicles, (i) 1 forward 0 left, (ii) 1 forward 1 left, (iii) 0 forward 1 left should all be free.

For 1 forward+right wrt the new positions of all vehicles, (i) 1 forward 0 right, (ii) 1 forward 1 right, (iii) 0 forward 1 right should all be free

Sample Input

```
1
3 20 3
0 8
1 5
2 13
1
```

Sample Output

```
11
```

SAMPLE INPUT	SAMPLE OUTPUT
1 3 20 3 0 8 1 5 2 13 1	11

Time Limit:	10.0 sec(s) for each input file.
Memory Limit:	5560 MB
Source Limit:	1024 KB
Marking Scheme:	Score is assigned if any testcase passes.
Allowed Languages:	C, C++, C++14, C++17, Java, Java 8, Java 14, Python, Python 3, Python 3.8

Question 4

Max. score: 5

This problem is no longer available for practice. Apology for any inconvenience!

In question 3 print the grid number at which the self-driving car should be after every time step. In case of a tie, a lexicographically smaller path is better (as integers 2<11).

Sample Input

```
1
3 20 3
0 8
1 5
2 13
1
```

Sample Output

```
1 0
1 2
1 4
1 6
2 7
2 9
2 11
2 13
2 15
2 17
2 19
```