AI Lab Test 2

1a. A modern building can be viewed as a 3D array of grids of *m* rows (numbered 0 to *m*-1), *n* columns (numbered 0 to *n*-1)and *h* floors (numbered 0 to *h*-1). From a grid you can take any of the 4-neighbouring grids at a unit time (or you walk at a speed of 1 block per unit time). You can only go up and down to access other floors using either of the staircases. Using the staircase you can go up at the rate of 1 floor in 2 units of time, and down at the rate of 1 floor in 1 units of time. In other words going 1 floor up takes 2 units of time and going down 1 floor takes 1 unit of time. You may use stairs to navigate within the same floor as a non-obstacle grid as well. Passing through stairs in the same floor as if they were non-obstacle regions is acceptable. Calculate the time required to reach a goal from a specific source.

Constraints: $0<m,n\leq 300$, $0<h\leq 50$

Input format
The first input is *T*, the number of test cases. For every test case, the first line is *m* (number of rows), *n* (number of columns) and *h* (number of floors). Thereafter details of all floors are given from index 0 (ground floor) to index *h*-1 (top floor). Each floor details has *m* rows of *n* columns each. A cell can be 0 (obstacle), 1 (free) or 2 (staircase). The staircases shall run through all the floors. The next line contains *q*, the number of queries. Each query is a source-goal pair for the same map. For every query, the next line contains the floor number, row number and column number of source and the next line contains the floor number, row number and column number of goal.

Output format
For every test case, 1 line printing the time to reach the goal. Print -1 if the source and goal are disconnected.

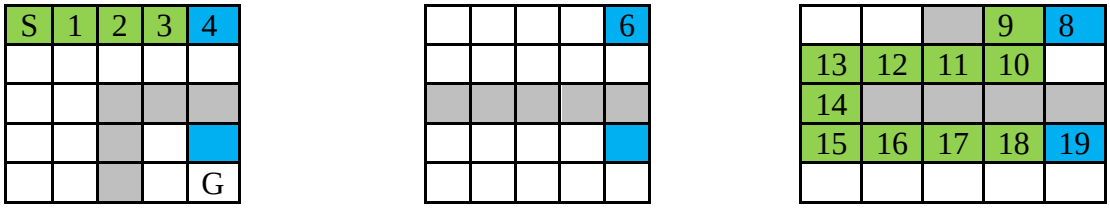
Sample Input
1 (number of test cases)
5 5 3 (m, n, h: double check the order of variables)
<ground 0^{th} floor details follow>
1 1 1 1 2
1 1 1 1 1
1 1 0 0 0
1 1 0 1 2
1 1 0 1 1
<1^{st} floor details follow>
1 1 1 1 2
1 1 1 1 1
0 0 0 0 0
1 1 1 1 2
1 1 1 1 1
<2^{nd} floor details follow>
1 1 0 1 2
1 1 1 1 1

1 0 0 0 0
1 1 1 1 2
1 1 1 1 1
1 (no. of queries)
0 0 0 (source for query 1, floor, row, column, double check the order of variables)
0 4 4 (goal for query 1, floor, row, column, double check the order of variables)

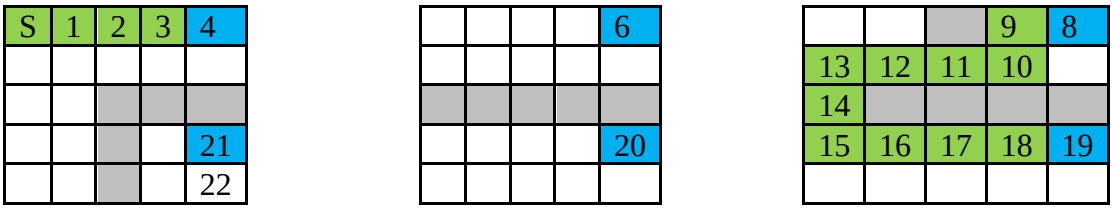
Sample Output
22

Explanation:
Legend: Gray (obstacle), White (free), Blue (Stairs), Green (Path), S (source), G (goal), Numbered (cost till step)

Ground Floor | **First Floor** | **Second Floor**



Continued
(read right to left)



1b. For question 1a, print the path instead of time. The path shall be printed as the sequence of coordinates (floor number <space> row number <space> column number) taken by the agent. In case of a tie, prefer the alphabetically smallest sequence. Print "NIL" if no path exists.

Sample Input: Same as Q1a

Sample Output (floor row column: double check the order of variables):
0 0 0
0 0 1
0 0 2
0 0 3
0 0 4

1 0 4
2 0 4
2 0 3
2 1 3
2 1 2
2 1 1
2 1 0
2 2 0
2 3 0
2 3 1
2 3 2
2 3 3
2 3 4
1 3 4
0 3 4
0 4 4

2a. In question 1 include lifts. The lifts are pre-programmed such that all lifts stop at all floors for a unit time, first go up from the ground floor, then return down from the top floor, and cycle in the same manner. Suppose there are 4 floors. All lifts are at floor 0 at time 0, floor 1 at time 1, floor 2 at time 2, floor 3 at time 3, floor 2 at time 4, floor 1 at time 5, floor 0 at time 6, floor 1 at time 7, floor 2 at time 8, and so on.

In order to catch a lift at coordinate (*x*,*y*) at time *t* as per the lift's visiting schedule, you must be at the neighboring coordinates of (*x*,*y*) at time *t*-1.

Moving in the same floor by 1 block takes 1 units of time and 2 units of effort (pain). Taking a stair up for 1 floor takes 2 unit of time and 4 units of effort (pain). Taking stairs down takes 1 unit of time and 3 units of effort (pain). Taking a lift up or down by a floor takes 1 unit of time and 2 unit of effort (pain). Waiting anywhere at any time has 1 unit of effort (pain).

You may use stairs and lifts to navigate within the same floor as a non-obstacle grid as well. Compute the least effort (pain) to reach a goal from a source.

Input format
The first input is *T*, the number of test cases. For every test case, the first line is *m* (number of rows), *n* (number of columns) and *h* (number of floors). Thereafter details of all floors are given from index 0 (ground floor) to index *h*-1 (top floor). Each floor details has *m* rows of *n* columns each. A cell can be 0 (obstacle), 1 (free), 2 (staircase) and 3 (lift). The staircases and lifts shall run through all the floors. The next line contains *q*, the number of query. Each query is a source-goal pair for the same map. For every query, the next line contains the floor, row number and column number of source and the next line contains the floor, row number and column number of goal.

Output format
For every test case, 1 line printing the effort to reach the goal. Print -1 if the source and goal are disconnected.

Sample Input (spaces for clarity)

```
1
5 5 3

1 3 1 1 2
1 1 1 1 1
1 1 0 0 0
1 1 0 1 3
1 1 0 1 1

1 3 1 1 2
1 1 1 1 1
0 0 0 0 0
1 1 1 1 3
1 1 1 1 1

1 3 0 1 2
1 1 1 1 1
1 0 0 0 0
1 1 1 1 3
1 1 1 1 1

1
0 0 0
0 4 4
```

Sample Output

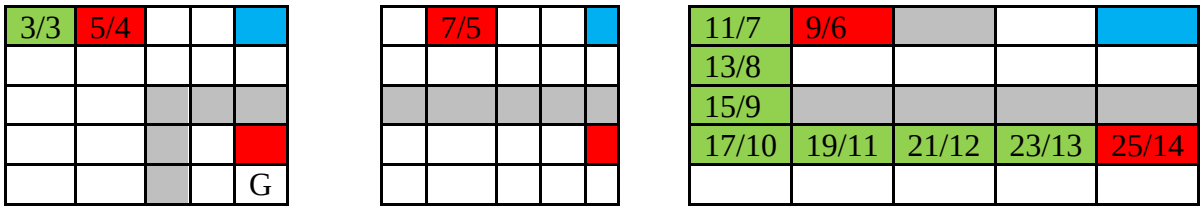
```
31
```

Explanation

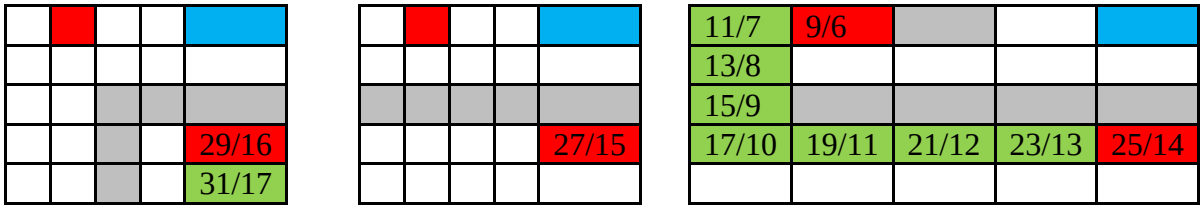
Legend: Red (Lift), Gray (obstacle), White (free), Blue (Stairs), Green (Path), S (source), G (goal), x/y (effort/time)

Note: Agent waits for 3 units of time at source, waiting for the lift to pick it up at t=4. Luckily there was no wait on the way down!

Ground Floor **First Floor** **Second Floor**



Continued
(read right to left)



2b. For question 2a, print the path instead of time. The path shall be printed as the sequence of coordinates taken by the agent. In case of a tie, prefer the alphabetically smallest sequence.

Sample Input: Same as question 2a

Sample Output

0 0 0
0 0 1
1 0 1
2 0 1
2 0 0
2 1 0
2 2 0
2 3 0
2 3 1
2 3 2
2 3 3
2 3 4
1 3 4
0 3 4
0 4 4