Validate paths.

Vocabulary

- Step: [N|E|S|W].

Given the current position (r,c), performing a step will lead to:

- N: (r-1,c)
- E: (r,c+1)
- S: (r+1,c)
- W:(r,c-1)

- Path: (c sp n s_1 s_2 … s_n), where *c* is the color of the path, *sp* is the starting position of the path, *n* is the length of the path, and s_i is the i-th step in the path.

For every path you can assume:

- It has a valid color: $1 <= c_{path} <= (pNr/2)$
- Starts from a point. $sp_{path} \in$ set of points, and $c_{path} == c_{sp}$

A path is invalid if any of the following cases occur:

1. Does not end in the corresponding place. (the second point with the same color)

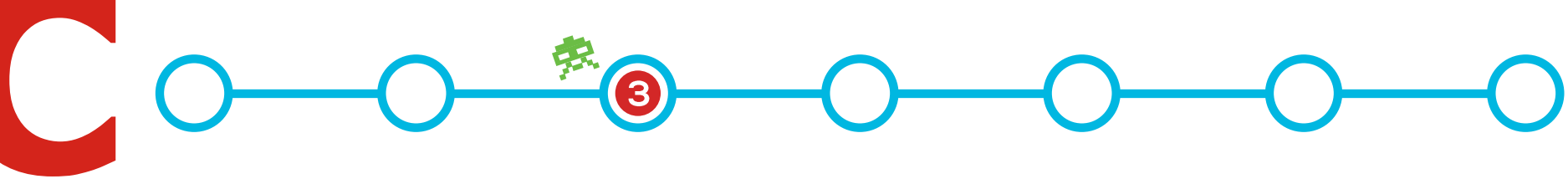
   a valid paths end point is :

   $ep_{path} \in$ set of points, and $c_{path} == c_{ep}$, where *ep* is the position after performing all the steps of the path
2. Goes out of bounds.
3. Crosses itself.
4. Touches point of a different color.

For each input specify if it's valid (code = 1) or invalid (code = -1).

Also specify the index of the step that caused the invalidity. For valid paths and paths that don't finish in the corresponding place, return the length of the path.

If there are multiple invalidities output the index of the first invalid step.



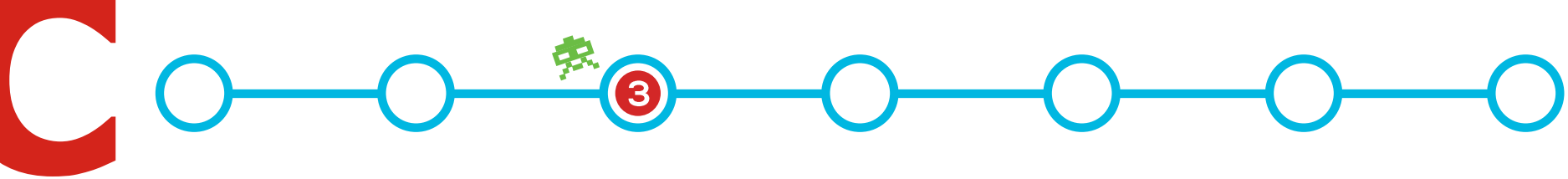
Input

rows cols numberOfPoints $Point_1$ $Point_2$ … $Point_{numberOfPoints}$ numberOfPaths $Path_1$ $Path_2$ … $Path_{numberOfPaths}$

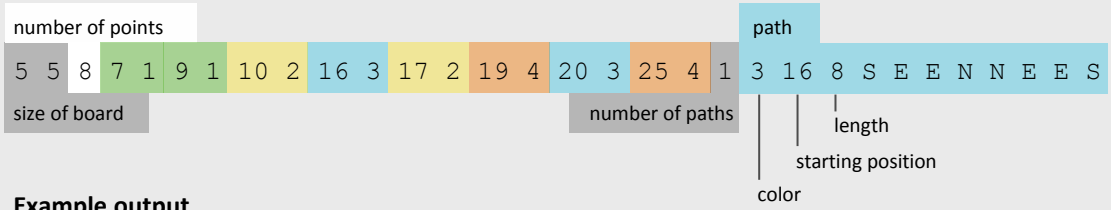
numberOfPaths will always be 1 (for this level)

Output

code index

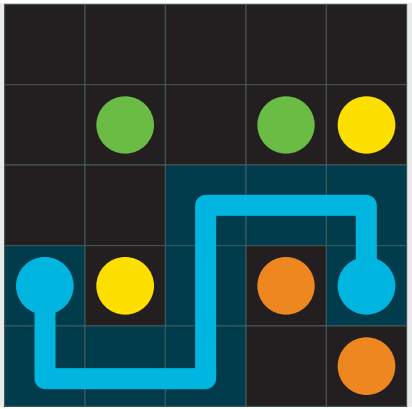


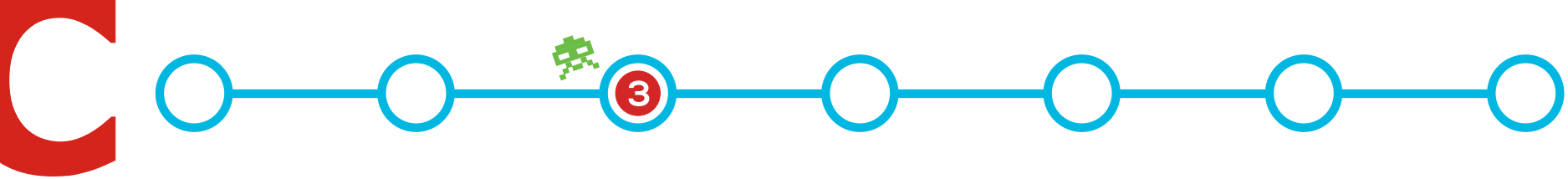
Example input



Example output

```
1 8
```



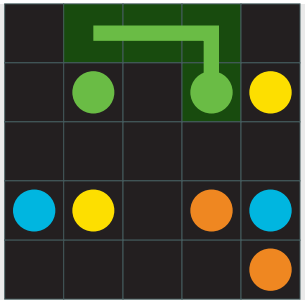


Example of invalidity 1

```
5 5 8 7 1 9 1 10 2 16 3 17 2 19 4 20 3 25 4 1 1 9 3 N W W
```

Example output

```
-1 3
```

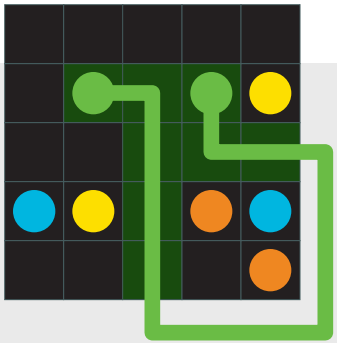


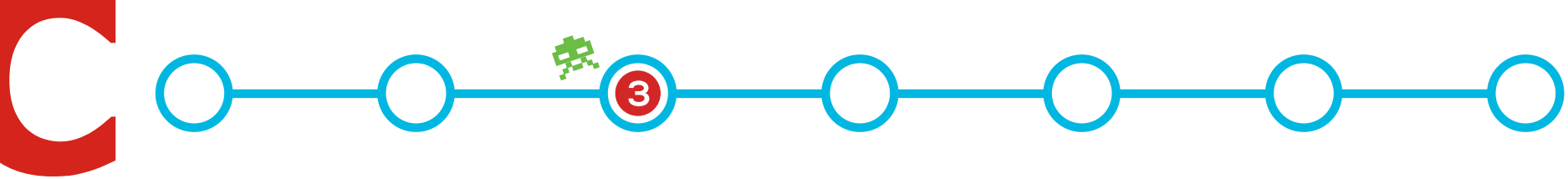
Example of invalidity 2

```
5 5 8 7 1 9 1 10 2 16 3 17 2 19 4 20 3 25 4 1 1 9 14 S E E S S S W W W
N N N N W
```

Example output

```
-1 3
```



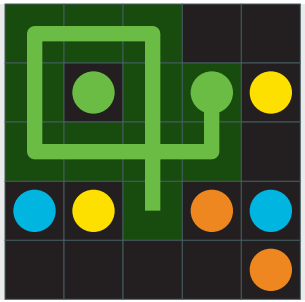


Example of invalidity 3

5 5 8 7 1 9 1 10 2 16 3 17 2 19 4 20 3 25 4 1 1 9 11 S W W W N N E E S S S

Example output

-1 10



Example of invalidity 4

5 5 8 7 1 9 1 10 2 16 3 17 2 19 4 20 3 25 4 1 1 9 3 S S W

Example output

-1 2