

# Snacks! (Easy)

Problem

Submissions

Leaderboard

Discussions

There's no one in the workshop who loves snacks as much as Anish. One day however, the juniors under the guidance of Dhruv decided to steal his snacks and scatter them all over in the hockey ground. Anish was angry at first, as he didn't want to go in search of his snacks midday, so he had an idea. He decided to use a drone to retrieve the snacks. He needs your help to find the best way to retrieve dem snacks!

In order to find out where they are located in the field, he takes the drone really high up and let your algorithm do its job. The algorithm first detects where all the snacks are located in the map, which Anish has already implemented. Now he wants you to find path to follow with which he can retrieve those snacks. Mind you, there are obstacles on the ground into which you don't want to crash into! Be sure to retrieve all the snacks as any snacks you leave out will be devoured by Dhruv :)

You will be given a map with the location of the snacks and obstacles in it. The map will be in the form of 20X20 matrix and you have to provide a path through it that goes through the cells which contain the snacks.

**NOTE:** All test cases will appear to be passed even if you don't output anything. We're using a custom checker, so please check your score after every submission to see how your algorithm performed.

NOTE: All test cases will appear to be passed even if you don't output anything. We're using a custom checker, so please check your score after every submission to see how your algorithm performed.

Explanation :

**\*\*Input Matrix: \*\***

2 2 2 0 2 0

2 2 0 1 2 0

0 0 0 2 0 1

2 0 1 0 1 2

2 2 2 2 2 2

2 2 2 2 2 2

**\*\*Output path: \*\***

3 2 2 3 4 5

1 2 3 3 3 2

NOTE: This explanation is only provided for you to understand the problem statement. The test cases will contain maps of 20x20.

Here, the first line denotes the x co-ordinates(column index) of the path taken while the next line denotes the y co-ordinates(row index). Hence, the path is (3,1),(2,2)... so on. The origin is the top left of the matrix

## Input Format

The candidate will be provided with  $n$  20X20 matrixes that are to be used as the map for the challenge. The 2s are the obstacles, 1s are the objects and 0s are the free space.

## Constraints

Obstacles = 2

Objects = 1

Free Space = 0

Note: Assume the obstacle heights are larger than the maximum altitude of the drone, hence you can't fly over them

## Output Format

You need to output two lines of numbers, the first one representing the x-coordinates of the path taken and the second line representing the y-coordinates.



### Sample Input 0

```
0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 1 0 0 0 1 0 0 0 0 1 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2
0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2
0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2
0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2
2 2 2 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2
2 2 2 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
2 2 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0
0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 0 0 0 0
0 0 0 0 2 2 2 2 0 0 0 2 2 2 2 2 0 0 0 0
0 0 0 1 2 2 2 2 0 0 0 2 2 2 2 2 0 0 0 0
0 0 0 0 2 2 2 2 1 0 0 0 0 0 1 1 0 0 0 0
0 0 0 0 2 2 2 2 0 0 1 0 0 0 0 0 0 0 0 0
0 0 0 0 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 2 2 2 2 0 0 1 0 0 0 1 0 0 0 0 0
```

### Sample Output 0

```
3 3 3 4 4 3 3 4 5 6 7 8 9 10 11 11 12 13 14 15 16 17 18 19 19 19 19 18 17 16 15 14 14 14 14
13 12 11 10 10 10 9 8 8 8 8 8 8 8 9 9 9 9 9 9 10 11 11 12 13 14 15 16 15 14 13 12 11 10 9
8 7 6 5 4 3 2
15 14 13 12 11 10 9 8 8 8 8 8 8 9 10 11 12 12 12 12 11 10 9 9 10 11 12 13 14 15 16 16 17 18
19 19 19 19 19 18 17 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 2 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 0 0 0 0
```

### Explanation 0

NOTE: There may be multiple optimal solutions to the problem. The sample example given here needn't be the only solution that gives maximum points.