Week 6 Graded Problem - Quiz 3

Topic: Dynamic Programming

Your adventure continues! This time you decided to find the **Deer Talisman** which is rumored to reside below Hacettepe University Faculty of Letters! However, the previous holder of this talisman was a master of **memoization** so you need to prepare accordingly.

The faculty building is famous for its complexity even among different solar systems. However, inside of the faculty building, there are guardians which prevent creatures that are aware of the talisman from getting near to it. So you decided to use the previous owner of the talisman's helper robot, **URobot** which will take the deer talisman for you.

You have dozen of maps of the building which include the possible location of the talisman. From your previous adventures of yours, you know the number of distinct paths that URobot can take from the entrance to the talisman. To find the correct map using this number, you decided to write a program that **given the map of the building, it calculates the number of distinct paths URobot can take**.

URobot's specifications are as follows:

(a) It can only move down and right in your map.

(b) It moves 1 meter at a time.

(c) It starts to traverse the map always at the position 0,0.

You are given an input file that contains both **the map** with 0 and 1's where 1's represent walls that URobot can't pass through and **the talisman's position (0 indexed)**. The Java code for reading the input file is already given to you, hence you are only expected to write the algorithm that returns the number of distinct paths given 2D array and the talisman's position (0 indexed).

Input

The inputs will be given as command line arguments to your Main class in the following format:

```
java Main <inputfile>
```

For example, to run your program on map.txt with the condition in Example Test Case #1

```
java Main map.txt
```

Output

You are expected to output the number of distinct paths that URobot can go from the starting point to the talisman's position to the STDOUT. For example, for the given sample run command and the sample map.txt in Example Test Case #1 your output should be:

```
4
```

If there is no path to the talisman that URobot can go, then simply output 0 to STDOUT.

Input File Organization

$n_{rows}\ n_{columns}$

$x_{0,0}\ x_{0,1}\ \dots\ x_{0,n_{columns}-1}$

$x_{1,0}\ x_{1,1}\ \dots\ x_{1,n_{columns}-1}$

$\dots$

$x_{n_r-1,0}\ x_{n_r-1,1}\ \dots\ x_{n_r-1,n_c-1}$

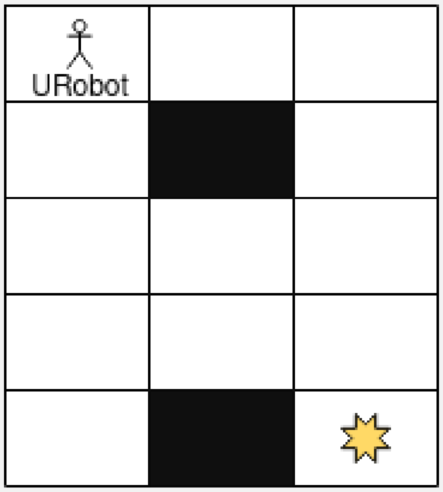
$Talisman_x\ Talisman_y$

Example

```
5 7
0 0 1 1 0 0 0
0 1 0 0 1 0 1
0 0 0 1 0 0 0
0 1 0 0 0 0 1
0 1 1 1 0 0 0
4 6
```

Important rules You MUST use this starter code. Do not change any function signatures in the given starter codes, those are given to ensure that you pass the unit tests in autograding. Only complete the TODOs.

Example Test Case #1



Distinct paths are in the following format: $x_1, y_1 \rightarrow x_2, y_2 \rightarrow ...$ where x_n, y_n corresponds to row index and column index in the array (starting from 0)

Input:

```
5 3
0 0 0
0 1 0
0 0 0
0 0 0
0 1 0
4 2
```

Distinct Paths:

(a) 0,0→0,1→0,2→1,2→2,2→3,2→4,2

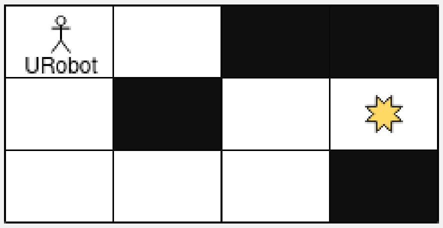
(b) 0,0→1,0→2,0→2,1→2,2→3,2→4,2

(c) 0,0→1,0→2,0→2,1→3,1→3,2→4,2

(d) 0,0→1,0→2,0→3,0→3,1→3,2→4,2

Output: 4

Example Test Case #2



Input:

```
3 4
0 0 1 1
0 1 0 0
0 0 0 1
1 3
```

Distinct Paths:
None
Explanation: Since URobot can't go up, it won't be able to reach the talisman's position
Output: 0