

Unique Paths II

Given a 'N' * 'M' maze with obstacles, count and return the number of unique paths to reach the right-bottom cell from the top-left cell. A cell in the given maze has a value '-1' if it is a blockage or dead-end, else 0. From a given cell, we are allowed to move to cells (i+1, j) and (i, j+1) only. Since the answer can be large, print it modulo $10^9 + 7$.

For Example :

Consider the maze below :

```
0 0 0
0 -1 0
0 0 0
```

There are two ways to reach the bottom left corner -

```
(1, 1) -> (1, 2) -> (1, 3) -> (2, 3) -> (3, 3)
(1, 1) -> (2, 1) -> (3, 1) -> (3, 2) -> (3, 3)
```

Hence the answer for the above test case is 2.

Input Format :

The first line contains a single integer 'T' denoting the number of test cases. Then each test case follows.

The first line of each test case contains integers 'N' and 'M' representing the size of the input grid.

Then 'N' lines follow, each of which contains 'M' space-separated integers denoting the elements of the matrix.

Output Format :

For each test case print an integer denoting the number of ways to reach the bottom right corner from the top-left corner modulo $10^9 + 7$.

Output for each test case will be printed in a separate line.

Note :

You are not required to print anything; it has already been taken care of. Just implement the function.

Constraints :

```
1 <= T <= 10
1 <= N,M <= 200
```

Note: It is guaranteed that the top-left cell does not have an obstacle.

Time Limit: 1 sec

Sample Input 1 :

```
2
```

```
2 2
0 0
0 0
3 3
0 0 0
0 -1 0
0 0 0
```

Sample Output 1 :

```
2
2
```

Explanation For Sample Output 1 :

For the first test case, there are two possible paths to reach (2, 2) from (1, 1) :

```
(1, 1) -> (1, 2) -> (2, 2)
(1, 1) -> (2, 1) -> (2, 2)
```

For the second test case, there are two ways to reach the bottom left corner -

```
(1, 1) -> (1, 2) -> (1, 3) -> (2, 3) -> (3, 3)
(1, 1) -> (2, 1) -> (3, 1) -> (3, 2) -> (3, 3)
```

Sample Input 2 :

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

Sample Output 2 :

```
0
```