Google BITWISE 2011 facebook

Repairing the Roof (Points: 300)

Mr. Bean has checked into his hotel in New Delhi. Unfortunately the hotel in which he is staying is under renovation and the roof is being repaired with new tiles. Mr. Bean decides to have a look himself and climbs to the roof top. He sees the whole roof infront of him, with several holes still remaining to be covered with tiles. The roof is an $\mathbf{M} \times \mathbf{N}$ rectangular region. Many parts have already been tiled (represented by 1), but there are several holes remaining (represented by 0). There cannot be a tile hanging within a hole (i.e. there cannot be a group of 1's surrounded by all 0's). The new tiles being used are of dimension (2×1) and (1×2) only. No other type of tile is available.

Mr. Bean suddenly decides to count the number of ways of finishing the job (i.e Covering the remaining holes with the given tiles). More the number of ways, faster the task will be completed. He soon realises that its not very easy, and needs your help. Help him by counting the number of ways of tiling the remaining holes using the given tiles only. Note that all holes must be covered completely and no two tiles can overlap.

Input Format:

First line contains the number of test cases **T**. For each test case, the first line of input will have two integers **M** and **N**. **M** represents the number of rows and **N** the number of columns in the roof plan. This will be followed by **M** lines of input. Each of these **M** lines will be a sequence of 0's and 1's of length **N**, separated by spaces. A 0 represents a missing space in the roof, while a 1 represents covered space of the roof.
Limits: $1 \leq \mathbf{T} \leq 10000$, $1 \leq \mathbf{M} \leq 30$, $1 \leq \mathbf{N} \leq 30$.

Output Format:

For each test case, output one integer giving the possible number ways of tiling the roof on a new line. If there is no possible way of tiling the roof, output 0. You can assume that the answer fits in `long long int` (64 bits).

Sample Input:

```
3

3 3
0 0 1
0 0 0
0 0 0

5 5
```

```
1 1 1 1 1
1 0 0 0 1
1 0 0 0 0
1 0 0 0 1
1 1 1 1 1

10 10
1 1 1 1 1 1 1 1 1 1
1 1 1 1 0 0 1 1 1 1
1 1 1 0 0 0 0 1 1 1
1 1 0 0 0 0 0 0 1 1
1 0 0 0 0 0 0 0 0 1
1 0 0 0 0 0 0 0 0 1
1 1 0 0 0 0 0 0 1 1
1 1 1 0 0 0 0 1 1 1
1 1 1 1 0 0 1 1 1 1
1 1 1 1 1 1 1 1 1 1
```

Sample Output:

```
4
0
1024
```

Instructions

- Your program should not print anything other than what is specified in the output format. A program with extraneous output (even a single space) will be treated as incorrect!
- While submitting your code, please select the language carefully *gcc/g++*. Using the wrong language will lead to compiler error.
- The only input/output functions allowed are `printf, scanf, cin, cout`. Perform all read/write operations through stdin/stdout. The solutions will be checked using command line redirection only.

Problem Setter: *Naveen Kr. Sharma*