

You are given a binary matrix (it contains only 0s and 1s) with dimensions 'N * M'. You need to find and return the sum of coverages of all zeros of the given matrix.

Coverage for a particular 0 is defined as the total number of '1s' around it (i.e., immediate left, immediate right, immediate up, and immediate bottom positions).

Input Format:

The first line of input contains an integer 'T' representing the number of the test cases. Then 'T' test cases follow.

The first line of each test case contains two space-separated integers, 'N', 'M', the dimensions of the matrix.

Next 'N' lines consist of 'M' space-separated integers denoting the matrix elements. Each element is either 0 or 1, as described in the problem statement.

Output Format:

For each test case, return the sum of coverage of all the 0s in the matrix.

Note:

You do not need to print anything; it has already been taken care of. Just implement the given function.

Constraints:

$$1 \leq T \leq 10$$

$$1 \leq N, M \leq 10^3$$

Time limit: 1 second

Sample Input 1:

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

Sample Output 1:

Explanation Of Input 1:

In the given matrix, there are 2 zeros.

For the 0 at the 1st row, 2nd column position, coverage is 2 (there is 1 adjacent top one and 1 adjacent right one).

For the 0 at the 2nd row, 2nd column the coverage is 2 (there is 1 adjacent top one and 1 adjacent right one).

Hence the net coverage is $2 + 2 = 4$.

Sample Input 2:

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

Sample Output 2

```
0
```