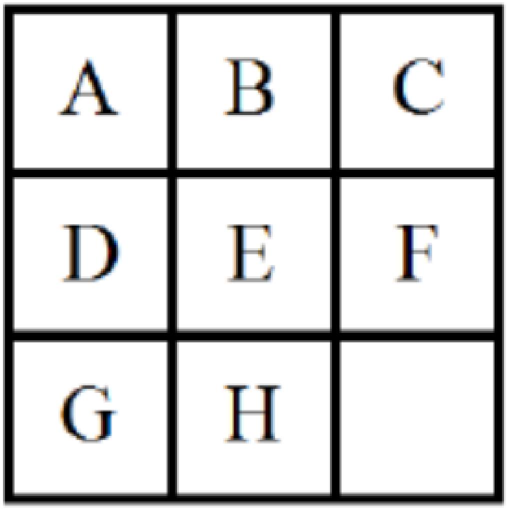
Contest Hall Preparation

Before the ACM-ACPC regional contest, the site director and the volunteers were very busy preparing for the contest. One of their tasks is to assign a table for each team such that no two teams from the same university are adjacent to each other. The site director decided not to waste his time doing this task and asked the judges to do it. The judges thought this could be a good problem to be used in the contest problems set. As they were very busy preparing for the contest, the judges decided to solve part of the problem and ask the contestants to solve the rest.

The judges will generate a number of layouts for the teams assignment to the tables and will ask the contestants to write a program to check whether each of these layouts is valid or not. If a layout is not valid the program should count how many different universities have at least two of their teams sitting adjacent to each other. "Well, you may use those solutions for the next year's contest", said by the chief judge *Ahmed Aly* to the site director.

The contest hall can be represented as a 2D grid of **N** rows with **M** cells in each row. Each cell in the grid is either occupied by a team or empty. There could be up to 8 teams adjacent to a single team. A team may have less than 8 adjacent teams if it is seated next to a hall edge or some of its adjacent cells are empty.

For example, in the layout shown in the following figure, team E has 7 adjacent teams, named A, B, C, D, F, G and H, while the adjacent teams to team A are B, D and E.



A	B	C
D	E	F
G	H	

Input

Your program will be tested on one or more test cases. The first line of the input will be a single integer **T**, the number of test cases (1 ≤ **T** ≤ 100). Followed by the test cases, each one starts with a line containing 2 integers separated by a single space **N** and **M** (1 ≤ **N**, **M** ≤ 100) representing the dimensions of the hall, followed by **N** lines each line contains **M** integers separated by a single space, representing the tables assignment in this row. Each integer represents the university ID of the team assigned to this table or '-1' if it is empty. All universities IDs are positive integers not greater than 100.

Output

For each test case, print on a single line integer, the number of different universities having at least two of their teams adjacent to each other.

Example

Sample Input

```
3
3 3
1 2 3
2 2 2
1 1 1
3 3
1 2 3
3 -1 1
2 -1 2
3 3
1 2 3
-1 1 5
1 2 4
```

Sample Output

```
2
0
1
```