You're the key programmer employed by a flight control center, and you've just been assigned a task which is to monitor the activity of aircrafts in a particular area.

Since some of the aircrafts in that area are so far away from the control that the sensors can't get a lock on them, it is impossible to determine the actual number of aircrafts in that area; however, the scanning of the flight control's sensor arrays has shown the amount of energy sensed in each grid of that area, and by analyzing that data, you will be able to get a general view of that area.

You have decided to first in order to write a program to simplify the calculation of the minimum number of aircrafts. you need to follow the following rules:

1. A grid from which the sensor arrays get no reading indicates a grid without any aircraft;
2. A grid with a positive value indicates a trace of an aircraft, and in that grid, a) An aircraft may be present, or b) It is the trace of exactly one aircraft traveling either horizontal or vertical (notice the aircraft can't change the flying direction halfway);
3. If a series of adjacent grids on a row or a column are the trace of one aircraft, then the amount of energy in them (from top to bottom or from left to right) must be either strictly increasing or strictly decreasing.

Input

There are multiple test cases in the input file. Each test case starts with two integers, N and M ($1 \leq N \leq 50$, $1 \leq M \leq 9$), the length and width of the area that the you're monitoring. Each of the following N lines consists of M integers, the j-th integer on the i-th line representing the amount of heat sensed by the flight control's sensor array. It is guaranteed that every integer in the input will fit into a 32-bit signed integer.

There is a blank line after each test case. A single line with $N = 0$ and $M = 0$ indicates the end of input file.

Output

For each test case, print one integer on a separate line, the minimum possible number of aircrafts, in the format as indicated in the sample output.

Sample Input

```
3 3
1 2 3
4 5 6
7 8 9

0 0
```

Sample Output

```
Case 1: 3
```