

6850 Hidden Plus Signs

In this problem, we want to find plus signs hidden in a number table. All we know about these plus signs and the number table are the followings.

- (1) The plus signs may overlap.
- (2) The width and height of each plus sign range from 3 to 11 units. For each plus sign, its width and height are equal.
- (3) One unit length equals to one table entry.
- (4) There are 2 to 9 plus signs in each table and no part of a plus sign lies outside a table.
- (5) The centers of plus signs do not lie inside another plus sign. In other words, the center of each sign does not overlap with any part of another plus sign.
- (6) The value of each table entry is the number of plus signs in the entry. For example, if there are 3 plus signs overlap in a table entry, the value in this entry is 3. If there is none, the value is 0.

For example, there are 2 plus signs in the following table. One sign has width of length 3 and another has width of length 5. The center of the first lies at row 2, column 2 of the number table, while the center of the second lies in row 3, column 3. The highlighted entries correspond to the centers of plus signs.

0	1	1	0	0
1	1	2	0	0
1	2	1	1	1
0	0	1	0	0
0	0	1	0	0

Your task is to write a program to count the number of plus signs in each table and report the position of the last plus sign's center. By last, it means the plus sign whose center lies lowest in a table. If there are multiple signs lie lowest, the center of the last plus sign is the rightmost among them.

Input

The first is the number of test cases T where $1 \leq T \leq 12$. For each test case:

1. The first line is the number of rows R and the number of columns C of the table where $3 \leq R, C \leq 30$.
2. The next R lines are the value of table entries from the top to bottom rows.

Output

For each test case, your program will write two output lines. The first is the number of plus signs hidden in a table. The second is the row and column of the last plus sign's center.

Note: Hint for Example Test Case 2

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

Sample Input

```

2
5 5
0 1 1 0 0
1 1 2 0 0
1 2 1 1 1
0 0 1 0 0
0 0 1 0 0
10 11
0 0 0 0 1 1 0 0 0 0 0
0 0 0 0 1 1 0 1 0 0 0
0 0 1 1 1 2 2 1 1 0 0
0 0 1 2 2 1 1 2 2 0 0
0 0 1 1 3 1 0 0 1 0 0
0 0 0 2 1 2 2 1 1 1 1
0 1 0 0 1 1 1 0 1 1 0
1 1 1 0 1 1 1 1 3 1 1
0 1 0 0 0 0 1 0 0 1 0
0 0 0 0 0 0 1 0 0 0 0

```

Sample Output

```

2
3 3
9
8 10

```