

Once upon a time there lived an old wise king. He had only one son. The prince had the idea that he (the prince) was about to be the future king; that's why he became too lazy day by day. This made the wise king a bit worried because his lazy son couldn't be the correct man for the throne. He couldn't sleep; his kingdom was in need of a perfect king, not the lazy king.

The old king grew older. One day, while doing his regular works, he found an excellent idea. The next day he called his son. He said the prince that he wanted the prince to have his own kingdom. The prince became excited with joy. The king continued that the prince can have a land from the king's kingdom, but he should start walking after sunrise and cover a rectangular area before sunset. Then the land would be given to the prince.

The lazy prince thought, "It would be an easy task!" That's why he wanted to find the maximum rectangular area in king's land. But there were some rocks in the kingdom, and the prince didn't want any rocks in his new land. He would rather take nothing, but no rocks.

The king's kingdom can be thought of an $m \times n$ grid, where m is the number of rows and n is the number of columns. Each cell in the grid is a small rectangular land whose area is 1. If a land contains rock it will be denoted by '1', otherwise it will be denoted by '0'. The prince can walk in the sides of the cells and he can either take a cell (land) or ignore it, but he can't take a part of a cell. Now your task is to find the maximum rectangular area the prince could cover.

Input

Input starts with in integer T (≤ 50) denoting number of cases.

Each case starts with a blank line. Next line contains two integers m and n ($1 \leq m, n \leq 2000$) denoting the rows and columns of the kingdom respectively.

Each of the next m lines will contain some integers in the following form

$k \ c \ p_1 \ p_2 \ p_3 \ \dots \ p_k$

meaning that in that row, the first p_1 columns are of type c , the next p_2 columns are of type 'opposite of c ', the next p_3 columns are of type c , the next p_4 columns are of type 'opposite of c ' and so on. You can assume that c will be either '0' or '1', and opposite of '0' is '1' and vice versa. You can also assume that $p_1 + p_2 + p_3 + \dots + p_k = n$ and p_i is positive.

Output

For each case, print the case number and the maximum area the prince could cover.

Notes:

- 1) For the first case the land covered by the prince is

```

0110110
0000010
1000001
0100001
1100010
    
```

- 2) For the second case the land covered by the prince is

```

001
100
101
    
```

Sample Input

```

2

5 7
5 0 1 2 1 2 1
3 0 5 1 1
3 1 1 5 1
4 0 1 1 4 1
4 1 2 3 1 1

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

Sample Output

```

Case 1: 12
Case 2: 3
    
```