C Christmas Presents

Author: Stefan Toman

Approximately 25% of the people who buy christmas presents begin to search for presents as late as one week before christmas. Lea has a different approach to this problem: She buys all the presents long before and uses December to decide which presents will be assigned to which of her friends. Lea has some constraints when she creates the assignment:

- All of Lea's friends should get a present.
- To be fair, each of her friends gets at most one present.
- None of the presents will be divided into multiple parts.
- Presents should only be assigned to a friend if he or she will like the present.

Luckily, Lea knows exactly which of her friends will like each present. She wants to prepare the algorithm to distribute the presents among her friends early before christmas, too. Can you help her writing it?

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with a line containing two integers n, the number of Lea's friends, and m, the number of presents Lea bought. The friends and presents are numbered from 1 to n or m, respectively. n lines follow describing the preferences of her friends: The i-th line contains a string with the numbers of the presents friend i likes. The numbers are comma-separated and may be given as sections where the first and last present are separated by a dash. For instance the string "1,10,3,5-8" represents presents 1, 3, 5, 6, 7, 8 and 10. Note that this string may also be empty.

Output

For each test case, output one line containing "Case #i: x" where i is its number, starting at 1, and x is "yes" if there is an assignment satisfying all constraints or "no" if there is no such assignment. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 250$

- $1 \leq m \leq 250$
- No presents will be mentioned several times per line of input.
- Sections of presents will always be given with the smaller index first.

Sample Data

Input

```
14
3 3
1
1-2
2,3

2 2
1
1

2 5
1-4
2-4

2 1
1
1

3 5
3-5
3-5
1

4 5
1,5,2-3
3-5
4-5
2-3

3 5
1,2,4-5
3-5
1,2-3

1 3
2

3 5
5,2-3,1
1-5
1-4

3 1
1
1
1

1 3
2,3

3 3
2-3
1
2-3,1

4 4
1-2
1-2
3-4,2
1-4

1 3
2-3,1
```

Output

```
Case #1: yes
Case #2: no
Case #3: yes
Case #4: no
Case #5: yes
Case #6: yes
Case #7: yes
Case #8: yes
Case #9: yes
Case #10: no
Case #11: yes
Case #12: yes
Case #13: yes
Case #14: yes
```