

Printing

Lea is not very fond of reading on her monitor, she prefers printed documents. However, madness has befallen all the printers! If you send them a print job, they do not print the whole job but randomly choose just one page that they print. Thus Lea has to print the same document multiple times until she can gather all the pages she needs.

Lea has already sent a number of print jobs to the different printers. Can you tell her if it is possible that she will find every page of the document she wanted to print at least once?

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 printers, and m , the number of pages in Lea's document. The printers and pages are numbered from 1 to n or m , respectively. n lines follow describing the print jobs Lea has sent: The i -th line contains a non-empty string with the numbers of the pages printer i was sent. The numbers are comma-separated and may be given as sections where the first and last pages are separated by a dash. For instance the string "1,10,3,5-8" represents pages 1, 3, 5, 6, 7, 8 and 10.

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 it is possible that every page of the document has been printed already, "no" otherwise. 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 page will be mentioned several times per line of input, in particular sections will not overlap.
- Sections of pages will always be given with the smaller index first.

Sample Input 1

```
5
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
```

Sample Output 1

```
Case #1: yes
Case #2: no
Case #3: no
Case #4: yes
Case #5: no
```

Sample Input 2

```
9
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
```

Sample Output 2

```
Case #1: no
Case #2: no
Case #3: no
Case #4: no
Case #5: yes
Case #6: no
Case #7: yes
Case #8: yes
Case #9: no
```