Problem 1.A Australian Ballots

Australian ballots require that the voter rank the candidates in order of choice. Initially only the first choices are counted and if one candidate receives more than 50% of the vote, that candidate is elected. If no candidate receives more than 50%, all candidates tied for the lowest number of votes are eliminated. Ballots ranking these candidates first are recounted in favour of their highest ranked candidate who has not been eliminated. This process continues [that is, the lowest candidate is eliminated and each ballot is counted in favour of its ranked non-eliminated candidate] until one candidate receives more than 50% of the vote or until all candidates are tied.

Input

The input begins with a single positive integer on a line by itself indicating the number of the cases following, each of them as described below. This line is followed by a blank line, and there is also a blank line between two consecutive inputs.

The first line of input is an integer $n \leq 20$ indicating the number of candidates. The next n lines consist of the names of the candidates in order. Names may be up to 80 characters in length and may contain any printable characters. Up to 1000 lines follow; each contains the contents of a ballot. That is, each contains the numbers from 1 to n in some order. The first number indicates the candidate of first choice; the second number indicates candidate of second choice, and so on.

Output

For each test case, the output must follow the description below. The outputs of two consecutive cases will be separated by a blank line.

The Output consists of either a single line containing the name of the winner or several lines containing the names of the candidates who tied.

Sample Input

```
1
3
John Doe
Jane Smith
Sirhan Sirhan
1 2 3
2 1 3
2 3 1
1 2 3
3 1 2
```

Sample Output

```
John Doe
```

Problem 1.B Interstellar Clocks

It is the 23rd century. Humans have finally colonized several nearby solar systems and dwell happily in their planets. Space travel is anyway very expensive and most inhabitants of extraterrestrial planets never again leave the surface of their planet in their lifetimes. Thus, upon arrival to an uninhabited planet, the colonizers adjusted the traditional Earth system of keeping track of time to the specific situation of that planet, in a way it would be practical for that planet and at the same time similar to what they were accustomed to.

In most planets, colonizers split the duration of a day into m hours (of duration similar to that of Earth's hours) and split each of those hours into n minutes. They read time from clocks in which the hour-hand moves continuously at a steady rate of one revolution per day, and the minute-hand moves continuously at a steady rate of one revolution per hour; the day begins at 0h : 0min at which time both hands are pointing upwards.

An interstellar clock manufacturer came up with a revolutionary clock design in which both the hour-hand and the minute-hand are the same size! They figured you could anyway tell the correct time from the clock. For instance, assume that in a 12-hour clock (like those on Earth) one of the hands is pointing upwards (to the 12) and the other hand is pointing to the 3: we can be sure it is 3 o’clock because it is impossible that the hour-hand is exactly pointing to the 12 and the minute-hand is not also pointing to the 12. Therefore we know it is the hour-hand that is pointing to the 3 and from this we can tell the time.

However, users from some planets soon complained they were having trouble telling the time from certain positions of the hands. The clock manufacturer has hired you to write a program to determine how many times during a certain time period confusion may arise from the position of the clock hands.

Input

The input consists of several test cases. The first line of each case contains three positive integers m, n and nq, where m and n represent the number of hours in a day and the number of minutes in an hour for a certain planet, and nq is the number of queries that follow. You may assume $m, n, nq \le 1000$. Each query is contained in a separate line and contains two times of the day, t1 and t2 , separated by a space. Each time of the day is given in the following format:

h:min num/den

where *h* is the hours, *min* is the minutes and *num/den* is a fraction < 1 which corresponds to a fraction of a minute. For example, in Earth time, '3:30 1/2' would correspond to 3 hours, thirty minutes and 30 seconds (half a minute). You may assume that $0 \le h < m$, $0 \le min < n$, $0 \le num, den \le 1000$ and that $t1 \le t2$ (this means that, within a day, *t1* does not happen after *t2*).

Input is terminated when $m = n = nq = 0$, this case should not be processed.

Output

For each test case print the header 'Planet i:', without quotes, where i indicates the test case number (beginning from 1) and then print, for each query, a line with a non-negative integer indicating the number of occasions in which confusion may arise from the position of the clock hands during the time interval (t_1 , t_2]. If there are an infinite number of such occasions, print 'Infinity', without quotes, instead of a number. Separate output from different test cases by a blank line.

Sample Input

```
12 60 2
11:0 0/1 11:0 0/1
6:45 1/3 7:1 2/5
0 0 0
```

Sample Output

```
Planet 1:
0
3
```

Problem 1.C Not That Kind of Graph

Your task is to graph the price of a stock over time. In one unit of time, the stock can either **R**ise, **F**all or stay **C**onstant. The stock's price will be given to you as a string of `R`'s, `F`'s and `C`'s. You need to graph it using the characters '`/`' (slash), '`\`' (backslash) and '`_`' (underscore).

Input

The first line of input gives the number of cases, N. N test cases follow. Each one contains a string of at least 1 and at most 50 upper case characters (`R`, `F` or `C`).

Output

For each test case, output the line '`Case #`x`:`', where x is the number of the test case. Then print the graph, as shown in the sample output, including the x- and y-axes. The x-axis should be one character longer than the graph, and there should be one space between the y-axis and the start of the graph. There should be no trailing spaces on any line. Do not print unnecessary lines. The x-axis should always appear directly below the graph. Finally, print an empty line after each test case.

Sample Input

```
1
RCRFCRFFCCRRC
```

Sample Output

```
Case #1:
|             _
|  _/\_/\    /
| /      \__/
+--------------
```

Problem 1.D How many 0's?

A Benedict monk No. 16 writes down the decimal representations of all natural numbers between and including m and n, m ≤ n. How many 0's will he write down?

Input

Input consists of a sequence of lines. Each line contains two unsigned 32-bit integers m and n, m ≤ n. The last line of input has the value of m negative and this line should not be processed.

Output

For each line of input print one line of output with one integer number giving the number of 0's written down by the monk.

Sample Input

```
10 11
100 200
0 500
1234567890 2345678901
0 4294967295
-1 -1
```

Sample Output

```
1
22
92
987654304
3825876150
```