Problem 1. Stable group

(Time Limit: 2 seconds)

Problem Description

There are several people. Some of them may dislike each other. If two people dislike each other are in a group, the group is unstable. Your task is to find a stable group of as many people as possible, that is, no one dislikes another in the group.

Input Format

The first line contains two integers m and n, in which m is the number of dislike pairs and n is the number of people. The people are labeled from 0 to n-1, and dislike pairs are given by the labels of two people, line by line. For example, the sample input contains 6 dislike pairs among 5 people. The pairs are (0,1), (1,2), (1,3), (1,4), (2,3), and (3,4). It is assumed that $1 < n < 101$.

Output Format

Output the maximum number of people of a stable group in one line. For the example, the maximum stable group is {0,2,4} which contains 3 people.

Example

Sample Input:	**Sample Output:**
6 5 0 1 1 2 1 3 1 4 2 3 3 4	3

Problem 2. Manufacturing a Spacecraft

(Time Limit: 3 seconds)

Problem Description

The space agency of the United Federation of Planets (UFP) is developing the new-generation spacecraft. Due to its modular design, the spacecraft comprises of a sequence of m components that can be manufactured independently. To assemble these components into the whole spacecraft, each component except the first and the last ones is needed to be connected to its predecessor and successor via an interface. To speed up the process of production, the manufacturing of these components is delegated to k manufactures ($k \leq m$). However, the interface between components produced by different manufactures may constitute a vulnerability point. Hence, to minimize the number of vulnerability points, every manufacturer must be responsible for a continuous sequence of components. Formally, there are m components (numbered 1, 2, ..., m) and it needs time t_i to finish the component i for $1 \leq i \leq m$. The space agency must find an increasing succession of numbers $0=n_0<n_1<n_2<...<n_{k-1}<n_k=m$ such that the j-th manufacturer is responsible for the production of the sequence of components with numbers between $n_{j-1}+1$ and n_j. The time that a manufacturer must spend to finish its work is the total time needed to manufacture all components assigned to it. The whole spacecraft is finished once the manufacturer that needs maximum time finishes its task. Therefore, our goal is to minimize the maximum time spent by a single manufacturer. Your task is to find the optimal time needed to manufacture the spacecraft.

Technical Specification

- The numbers of manufacturers and components satisfy $1 \leq k \leq m \leq 500$.
- The time needed for the production of a component satisfies $0 < t_i \leq 10^7$.

Input Format

The first line is an integer which indicates the number of test cases. Each case consists of exactly two lines. At the first line, there are two integers m and k, which indicate m components and k manufactures. At the second line, there are m integers t_1, t_2, ..., t_m separated by spaces.

Output Format

For each case, print exactly one line. The line must contain a positive integer denoting the minimum time that is needed to manufacture the spacecraft.

Example

Sample Input:	**Sample Output:**
2 9 3 100 200 300 400 500 600 700 800 900 5 4 100 100 100 100 100	1700 200

Problem 3. Prepare to be Moon-Struck!

(Time Limit: 5 second)

Problem Description

Elune, the goddess of moon, has a row of *N* moonstones set in front of her palace all the way to the entrance. These stones, like moon, can turn dark or light under the goddess's influence. Upon full moon, Elune will set all the moonstones to light. But as Elune is the goddess of moon, her mood and preference change quickly from time to time. She likes to modify some of the moonstones every time she returns to her temple and passes through them. For whatever strange moon goddess reason, she must start modifying them by turning one of the light moonstones into dark. After that, she can modify the rest of the stones however she wants. And as an important goddess as she is, she is busy all the time so she cannot go backwards to modify the ones she has already passed through. Note that she only modifies the moonstones when she returns to the temple, never when she is leaving it. Suppose there are 15 moonstones and their pattern is:

Entrance ①②③❹❺⑥⑦⑧❾❿⓫⑫⑬⑭⓯ Elune's palace

where white circles represent light moonstones and black circles represent dark ones. Elune may change the pattern above into:

Entrance ①②③❹❺❻⑦⑧❾⑩⑪⓬⓭⑭⓯ Elune's palace

But she will NOT change the first mentioned pattern into

Entrance ①②③❹⑤❻⑦⑧❾⑩⑪⓬⓭⑭⓯ Elune's palace

Of course, those mortal beings cannot see her as she comes and goes. But they can, however, see the changing of the moonstones. The high priestess loves to observe and record the changes as she thinks it might be some omen that the goddess is giving to her. How silly of her, the moon goddess is just doing that for fun!

Recently, there has been a series of criminal activities near the Temple of Elune. The Ancients that witnessed the crimes are so old that they cannot remember the time it happened. They are so old that time doesn't make sense to them anymore, you see. Luckily, they do remember the pattern of the moonstones when the crime happened! That combined with the high priestess's record, there may be hope for us to piece out the time the crime was witnessed.

Input Format

The first line in the input is the number of test cases *T*, which will not exceed 10. The first line of each test case contains three integers *N*, *M* and *Q*. There are *N* moonstones, the high priestess recorded M patterns in the order that Elune may modify them, and there are Q criminal activities that happened. It is guaranteed that N will not exceed 100000, M will not exceed 1000, and Q will not exceed 500.

The next M lines are the patterns the high priestess recorded and the Q lines after that are the patterns that the Ancients saw during the crime. These patterns are described by a run-length code “a r_1 r_2 ... r_a” which represents there are a runs, the first run is r_1 of light moonstones, the second run is r_2 of dark moonstones, and so on. For instance, “4 4 3 2 1” represents ①②③④❺❻❼⑧⑨❿, and “5 0 1 2 3 4” represents ❶②③❹❺❻⑦⑧⑨⑩. It is guaranteed that only r_1 can be zero, $r_1+r_2+...+r_a = N$ and a will not exceed 1000.

Note: The input is quite large. `cin` might be too slow.

Output Format

For each criminal activity, output one line:

1. If it is before the first recorded pattern, output “<1”.
2. If it is after the M-th recorded pattern, output “>M”.
3. If it is exactly the i-th recorded pattern, output “i”.
4. If it is between the i-th and the (i+1)-th recorded pattern, output “i i+1”.

Example

Sample Input:	Sample Output:
2 3 2 1 1 3 2 1 2 3 1 1 1 10 3 5 3 4 2 4 4 2 3 3 2 5 0 1 2 3 4 2 2 8 2 1 9 3 4 2 4 2 0 10 1 10	1 2 2 3 2 3 1 >3 <1

Problem 4. Common Gene

(Time Limit: 4 seconds)

Problem Description

A gene is a sequence of meaningful part of DNA sequence. In order to test the similarity of two creatures, we want to compare their DNA sequence. If any two creatures are of the same family, we know that their DNA sequences are at least 99.5% alike. That is, their longest common subsequence (LCS) has greater or equal to 99.5% of their original lengths.

Given two DNA sequences (a string composed of four types of characters A, C, T and G). You are asked to report their similarity, or saying that they are of different families.

Technical Specification

- Number of test cases T, $1 \le T \le 26$
- A positive integer n, $1 \le n \le 50000$
- All input strings are of length n

Input Format

The first line of the input file contains an integer indicating the number of test data. Each test data contains n, and then two strings A and B.

Output Format

For each test case, please output "`different`" if the longest common subsequence is strictly less than $0.995n$. Otherwise, please output the length of the LCS over the original length. No need to reduce the fraction.

Example

Sample Input:	Sample Output:
3 20 ACTGACTGACTGACTGACTG ACTGACTGACTGACTGACGT 10 AAAAAAAAAA AAAAAAAAAA 10 ACAAAAAAAC ACCCCCCCCA	different 10/10 different

Problem 5. Logo Design

(Time Limit: 5 seconds)

Problem Description

A logo is a graphic mark or symbol commonly used by commercial enterprises, organizations and even individuals to aid promotion of public recognition. There are so many logos around, such as business logos, website logos, …, etc. Logos play an important role in our life.

Teddy is a famous logo designer. Recently, Teddy has encountered the folloiwng problem while designing a new logo. First, he draw a set S of n non-vertical straight-line segments in the plane. The segments are regarded as opaque obstacles, and their *upper envelope*, denoted by $U(S)$, consists of the portion of the segments visible from $y = +\infty$. For example, consider the set $S = \{s_1, s_2, s_3, s_4\}$ in Figure 1. For simplicity, let (x_1, y_1, x_2, y_2) denote the segment defined by the two endpoints (x_1, y_1) and (x_2, y_2). Then, s_1, s_2, s_3, s_4 are denoted, respectively, by (0, 5, 3, 10), (1, 8, 4, 1), (5, 4, 9, 4), and (6, 2, 10, 7). Point (6, 4) of s_3 is on the upper envelope, whereas point (6, 2) of s_4 is not. Figure 2 shows the upper envelope of S.

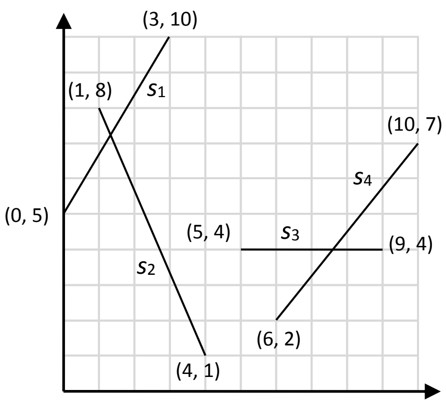

Figure 1. S

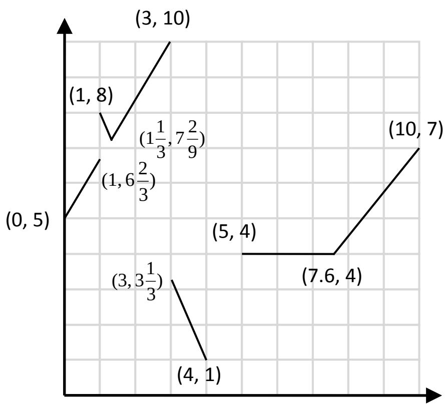

Figure 2. $U(S)$

A line segment p is called a *piece* of the upper envelope $U(S)$ if p is on $U(S)$ and no other line segment p' containing p is also on $U(S)$. For example, in Figure 2, there are six pieces: (0, 5, 1, $6\frac{2}{3}$), (1, 8, $1\frac{1}{3}$, $7\frac{2}{9}$), ($1\frac{1}{3}$, $7\frac{2}{9}$, 3, 10), (3, $3\frac{1}{3}$, 4, 1), (5, 4, 7.6, 4), (7.6, 4, 10, 7).

Teddy would like to known what is the number of pieces of the upper envelope $U(S)$. Please write a program to help Teddy.

Technical Specification

- The number of line segments n is an integer between 1 and 10^4.
- The x, y coordinates of the endpoints of each line segment are integers between 0 and 10^{15}.
- It is guaranteed that:
  (1) There is no vertical line segment, and
  (2) No two line segments induce a single line segment. We say that two line segments induce a single line segment if the union of their points represents a line segment. For example, (0, 3, 2, 1) and (2, 1, 3, 0) induce a single line segment (0, 3, 3, 0).

Input Format

The first line contains an integer $t \le 20$ indicating the number of test cases. The first line of each test case contains an integer n, which is the number of line segments. In the following n lines, the i-th line contains four integers x_i, y_i, x'_i, y'_i ($x_i < x'_i$), where (x_i, y_i) and (x'_i, y'_i) are the endpoints of the i-th line segment.

Output Format

For each test case, output the number of pieces of the upper envelope.

Example

Sample Input:	**Sample Output:**
2 4 0 5 3 10 1 8 4 1 5 4 9 4 6 2 10 7 3 1 0 999999999999999 1000000000000000 0 1 1000000000000000 999999999999999 499999999999999 500000000000000 500000000000001 500000000000000	6 4