Problem 1. Arithmetic Expression

(Time Limit: 1 second)

Problem Description

An arithmetic expression consists of integer numbers, basic operators such as addition, subtraction, multiplication, division, and parentheses. For example, the following arithmetic expressions are legal:

12 + 34 – 25 * 27
(12 + 34) – (25 * 27)
30 / (12 + 345) – 30 * (2 + 4)

And, the following arithmetic expressions are illegal:
+ 23 – 45 *
(12 – 34) * (15 – 5
) 23 – 14 * (23/34)

Please write a program to judge whether an input expression is legal or not.

Input Format

The first line contains an integer N (1≤N≤10) indicating the number of the test cases. For each test case, there is one string with the length less than 500 representing an arithmetic expression in one line.

Output Format

For each test case, if the expression is legal then output “legal expression”; otherwise output “illegal expression”.

Example

Sample Input:	**Sample Output:**
2 ((12-23)*(12+54))+(20-3) 12+3-20*20/	legal expression illegal expression

Problem 2. Exam Hints

(Time Limit: 1 second)

Problem Description

The final exam is coming!

According to the past experiences, the final exam would be ridiculously hard. There is some mysterious hint given by the teacher. The hint is a set of problem ids that will **DEFINITELY** help you to pass the exam.

The hint is given in the form like "1,2~3,5,8~8", which means 5 problem ids: 1,2,3,5,8. In other words, there are some problem range(s) separated by commas, and each problem range may be either a single number like "5", or two numbers with a tilde like "2~3". Note that the problem ranges in the hint may be overlapped.

Please list all the problem ids in the hint (and pass the exam).

Technical Specification

- The number of test case $T \leq 500$
- The hint will be valid, and contains only digits, commas, and tildes
- $1 \leq$ The number of problem ranges in a hint ≤ 500
- All problem ids are in the interval [1, 100]
- $L \leq R$ for all problem ranges with the form "*L*~*R*"

Input Format

The first line contains an integer *T* indicating the number of the test cases. For each test case, there is a hint of the final exam in one line.

Output Format

For each test case, output the number of problem ids and list all the problem ids in the hint in ascending order. Please follow the exactly same format of sample output ("N: id_1 id_2 … id_N").

Example

Sample Input:	**Sample Output:**
2 1,2~3,5,8~8 4~7,2,6~8,5	5: 1 2 3 5 8 6: 2 4 5 6 7 8

Problem 3. Super Go

(Time Limit: 1 second)

Problem Description

Go is a strategy board game for two players, and the aim is to surround more territory than the opponent. Go was considered to be a very challenging computational problem, because it possesses more possibilities than the total number of atoms in the visible universe. However, after several Go players were defeated by Google's AlphaGo program, the international Go community decided to create a harder version called Super Go. The original Go was played on a board only with 19*19 grid of lines. The Super Go is played on a board with 65535*65535 grid of lines. In addition, the original Go required complete surrounding of stones to claim a territory. The Super Go allows discontinuous surrounding of stones, and the claimed territory is measured by the minimum area of outermost surrounding stones (i.e., a smallest convex hull that enclosed all stones). Suppose the convex hull is composed of k stones: s_0, s_1, …, s_k where s_0=s_k and the i-th stone has coordinate (x_i,y_i), for i = 0 to k. The area of the convex hull is defined as $\left|1/2\sum_{i=1}^{k}(x_{i-1} * y_i) - (x_i * y_{i-1})\right|$. Given a set of stones, you are asked to write a program for finding the convex hull.

Technical Specification

- A positive integer n, $3 \le n \le 10000$, representing the number of stones.
- The coordinate of each stone (x,y) are integers and has the following range: $0 \le x,y < 65535$.

Input Format

The first line of the input file contains an integer indicating the number of test cases. Each test case starts with a line containing the number of stones n. The following n lines contain the (x,y) coordinate separated by a white space.

Output Format

For each test case, output the area of the claimed territory, which should be printed with two decimal places.

Example

Sample Input:	**Sample Output:**

2
5
0 0
1 1
2 0
0 2
2 2
3
0 0
3 0
0 1
	4.00
	1.50

Problem 4. Reb Lang

(Time Limit: 1 second)

Problem Description

Rebecca has designed a visualization programming language (VPL) called Reb-Lang, which has two types of elements: variable and function. This system let students program by linking variables and functions in a GUI environment. For example, x + y in Reb-Lang should be linked like Fig. 1.

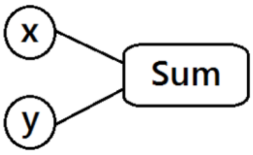

Figure 1: x + y in Reb-Lang.

In Reb-Lang's syntax, there are 3 rules:

1. A variable can link to one or more functions.
2. A function can have zero or multiple input ports which should be linked to a variable or an output port of a function.
3. The output port of a function can be linked to zero or multiple functions.

For example, (x + y) * y may look as in Fig. 2.

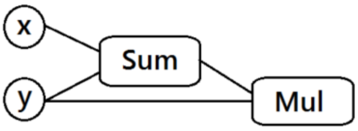

Figure 2: (x + y) * y in Reb-Lang.

In addition, there is a limitation: No backward links. A backward link means an output of a function is linked backward and become the input source of the same function (see examples in Fig. 3. Neither case1 nor case2 are allowed). Since backward links may cause infinite calculation, Rebecca decides to forbid these links.

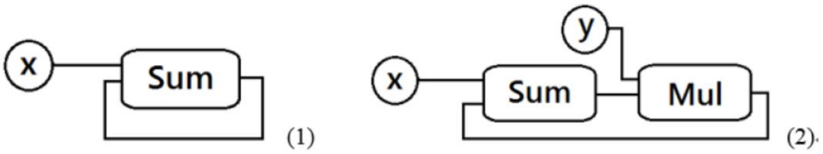

Figure 3: Two backward link examples.

Input Format

The test data begins with an integer k which is the number of test cases. In each test case, first line of input contains two integer, N and M, which stands for the number of element(s) and the number of link(s), $1 \leq N \leq 10^4$; $0 \leq M \leq 10^5$. Each of the next M lines contains two integer A and B, which means there is a link from element A (variable or function) to element B, where $1 \leq A \leq N$; $1 \leq B \leq N$.

Output Format

If there are any backward links, print “INVALID”. Print “VALID”, otherwise.

Example

Sample Input:	**Sample Output:**

1
4 5
1 3
2 3
2 4
3 4
4 3
	INVALID

Problem 5. Tollbooth Construction

(Time Limit: 3 seconds)

Problem Description

Kitten is a country with *n* cities. After few months of construction, the government totally built *n*-1 highways which connect these cities. Since then, the citizens can travel from any city *i* to other city *j* by driving their cars through one or more highways.

One day, the government noticed that these highways need to be maintained. Meow, the president of Kitten, decided to build some tollbooths to charge money along the highways. However, charging too much money would make the citizens unhappy. Thus, President Meow asked you to make a construction plan that will not cause the citizens unhappy. A citizen will be unhappy if he/she travels with charged 2 or more times. That is, there cannot have more than one tollbooth along the highways from any city *i* to the other city *j*.

Starting from a given city, there is at least one possible solution, because no tollbooth constructed is a trivial solution. In order to estimate the complexity of the problem, President Meow wants you to figure out the sum of all possible solutions starting from each of cities. For example, there are 4 possible solutions starting from city 1 in Figure 1. Similarly, there are 3 possible solutions starting from each of city 2 and city 3, as shown in Figure 2. Thus, the number of solutions starting from city 1, 2 and 3 are 4, 3 and 3, respectively. The sum of all possible solutions is equal to 10 = 4 + 3 + 3.

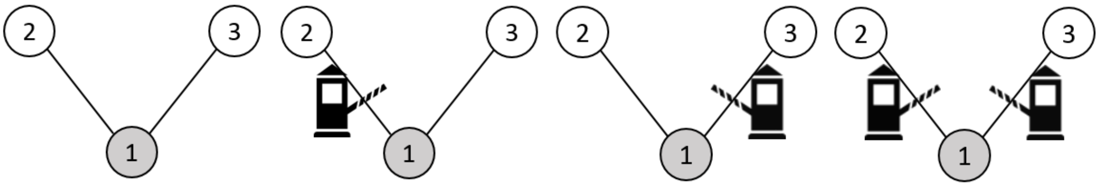

Figure 1

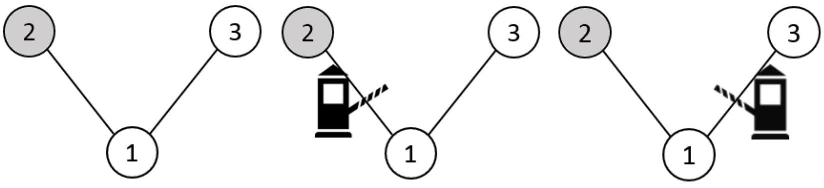

Figure 2

Technical Specification

- Number of test cases *T*, $1 \le T \le 10$.
- Number of cities *n*, $2 \le n \le 50000$.
- All cities are numbered from 1 to *n*.

Input Format

The first line of input contains an integer *t*, denoting the number of test cases. Each of the following test cases starts with a number *n*, denoting the number of cities. The next line contains *n*-1 integers C_2 C_3 C_4 … C_n, separated by single spaces, where C_i means that there is a highway connecting city *i* and city C_i, where $1 \le C_i \le$ i-1.

Output Format

For each test case, output a single integer *S* in a line, denoting the sum of all possible solutions starting from each of the cities, modulo by 1000000007 (10^9+7). For example, if the number of possible solutions starting from city *i* is S_i, then *S* equals to (S_1+S_2+…+S_n) modulo (10^9+7).

Example

Sample Input:	**Sample Output:**

2
3
1 1
5
1 2 3 4
	10
	35