A. Binomial coefficients

Gunnar is quite an old and forgetful researcher. Right now he is writing a paper on security in social networks and it actually involves some combinatorics. He wrote a program for calculating binomial coefficients to help him check some of his calculations.

A binomial coefficient is a number

$$\binom{n}{k} = \frac{n!}{k!(n-k)!},$$

where n and k are non-negative integers.

Gunnar used his program to calculate $\binom{n}{k}$ and got a number m as a result. Unfortunately, since he is forgetful, he forgot the numbers n and k he used as input. These two numbers were a result of a long calculation and they are written on one of many papers lying on his desk. Instead of trying to search for the papers, he tried to reconstruct the numbers n, k from the output he got. Can you help him and find all possible candidates?

Input

On the first line a positive integer: the number of test cases, at most 100. After that per test case:

- one line with an integer m ($2 \leq m \leq 10^{15}$): the output of Gunnar's program.

Output

Per test case:

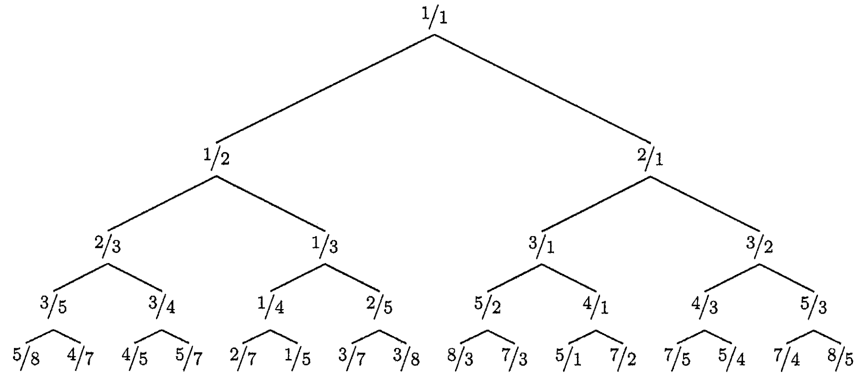
- one line with an integer: the number of ways of expressing m as a binomial coefficient.
- one line with all pairs (n, k) that satisfy $\binom{n}{k} = m$. Order them in increasing order of n and, in case of a tie, order them in increasing order of k. Format them as in the sample output.

Sample in- and output

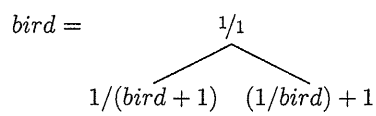
Input	Output
2 2 15	1 (2,1) 4 (6,2) (6,4) (15,1) (15,14)

B. Bird tree

The Bird tree¹ is an infinite binary tree, whose first 5 levels look as follows:



It can be defined as follows:



This is a *co-recursive* definition in which both occurrences of *bird* refer to the full (infinite) tree. The expression $bird + 1$ means that 1 is added to every fraction in the tree, and $1/bird$ means that every fraction in the tree is inverted (so a/b becomes b/a).

Surprisingly, the tree contains every positive rational number exactly once, so every reduced fraction is at a unique place in the tree. Hence, we can also describe a rational number by giving directions (`L` for left subtree, `R` for right subtree) in the Bird tree. For example, 2/5 is represented by `LRR`. Given a reduced fraction, return a string consisting of `L`'s and `R`'s: the directions to locate this fraction from the top of the tree.

Input

On the first line a positive integer: the number of test cases, at most 100. After that per test case:

- one line with two integers a and b ($1 \leq a, b \leq 10^9$), separated by a `'/'`. These represent the numerator and denominator of a reduced fraction. The integers a and b are not both equal to 1, and they satisfy $\gcd(a, b) = 1$.

For every test case the length of the string with directions will be at most 10 000.

Output

Per test case:

- one line with the string representation of the location of this fraction in the Bird tree.

[1] Hinze, R. (2009). The Bird tree. *J. Funct. Program.*, 19:491–508.

Sample in- and output

Input	Output
3 1/2 2/5 7/3	L LRR RLLR

C. Movie collection

Mr. K. I. has a very big movie collection. He has organized his collection in a big stack. Whenever he wants to watch one of the movies, he locates the movie in this stack and removes it carefully, ensuring that the stack doesn't fall over. After he finishes watching the movie, he places it at the top of the stack.

Since the stack of movies is so big, he needs to keep track of the position of each movie. It is sufficient to know for each movie how many movies are placed above it, since, with this information, its position in the stack can be calculated. Each movie is identified by a number printed on the movie box.

Your task is to implement a program which will keep track of the position of each movie. In particular, each time Mr. K. I. removes a movie box from the stack, your program should print the number of movies that were placed above it before it was removed.

Input

On the first line a positive integer: the number of test cases, at most 100. After that per test case:

- one line with two integers n and m ($1 \leq n, m \leq 100\,000$): the number of movies in the stack and the number of locate requests.
- one line with m integers $a_1, \ldots, a_m$ ($1 \leq a_i \leq n$) representing the identification numbers of movies that Mr. K. I. wants to watch.

For simplicity, assume the initial stack contains the movies with identification numbers $1, 2, \ldots, n$ in increasing order, where the movie box with label 1 is the top-most box.

Output

Per test case:

- one line with m integers, where the i-th integer gives the number of movie boxes above the box with label a_i, immediately before this box is removed from the stack.

Note that after each locate request a_i, the movie box with label a_i is placed at the top of the stack.

Sample in- and output

Input	Output
2 3 3 3 1 1 5 3 4 4 5	2 1 0 3 0 4

Problem D. Miscalculation

Input: Standard Input
Time Limit: 1 second

Bob is an elementary schoolboy, not so good at mathematics. He found Father's calculator and tried cheating on his homework using it. His homework was calculating given expressions containing multiplications and additions. Multiplications should be done prior to additions, of course, but the calculator evaluates the expression from left to right, neglecting the operator precedence. So his answers may be the result of either of the following two calculation rules.

- Doing multiplication before addition
- Doing calculation from left to right neglecting the operator precedence

Write a program that tells which of the rules is applied from an expression and his answer.

An expression consists of integers and operators. All the integers have only one digit, from 0 to 9. There are two kinds of operators + and *, which represent addition and multiplication, respectively.

The following is an example expression.

```
1+2*3+4
```

Calculating this expression with the multiplication-first rule, the answer is 11, as in Sample Input 1. With the left-to-right rule, however, the answer will be 13 as shown in Sample Input 2.

There may be cases in which both rules lead to the same result and you cannot tell which of the rules is applied. Moreover, Bob sometimes commits miscalculations. When neither rules would result in Bob's answer, it is clear that he actually did.

Input

The input consists of a single test case specified with two lines. The first line contains the expression to be calculated. The number of characters of the expression is always odd and less than or equal to 17. Each of the odd-numbered characters in the expression is a digit from '0' to '9'. Each of the even-numbered characters is an operator '+' or '*'. The second line contains an integer which ranges from 0 to 999999999, inclusive. This integer represents Bob's answer for the expression given in the first line.

Output

Output one of the following four characters:

- `M` When only the multiplication-first rule results Bob's answer.
- `L` When only the left-to-right rule results Bob's answer.
- `U` When both of the rules result Bob's answer.
- `I` When neither of the rules results Bob's answer.

Sample Input 1	Sample Output 1
1+2*3+4 11	M

Sample Input 2	Sample Output 2
1+2*3+4 13	L

Sample Input 3	Sample Output 3
3 3	U

Sample Input 4	Sample Output 4
1+2*3+4 9	I

Problem E.
Shopping

Input: Standard Input
Time Limit: 1 second

Your friend will enjoy shopping. She will walk through a mall along a straight street, where N individual shops (numbered from 1 to N) are aligned at regular intervals. Each shop has one door and is located at the one side of the street. The distances between the doors of the adjacent shops are the same length, i.e. a unit length. Starting shopping at the entrance of the mall, she visits shops in order to purchase goods. She has to go to the exit of the mall after shopping.

She requires some restrictions on visiting order of shops. Each of the restrictions indicates that she shall visit a shop before visiting another shop. For example, when she wants to buy a nice dress before choosing heels, she shall visit a boutique before visiting a shoe store. When the boutique is farther than the shoe store, she must pass the shoe store before visiting the boutique, and go back to the shoe store after visiting the boutique.

If only the order of the visiting shops satisfies all the restrictions, she can visit other shops in any order she likes.

Write a program to determine the minimum required walking length for her to move from the entrance to the exit.

Assume that the position of the door of the shop numbered k is k units far from the entrance, where the position of the exit is $N + 1$ units far from the entrance.

Input

The input consists of a single test case.

$$
\begin{array}{l}
N\ m \\
c_1\ d_1 \\
\vdots \\
c_m\ d_m
\end{array}
$$

The first line contains two integers N and m, where N $(1 \leq N \leq 1000)$ is the number of shops, and m $(0 \leq m \leq 500)$ is the number of restrictions. Each of the next m lines contains two integers c_i and d_i $(1 \leq c_i < d_i \leq N)$ indicating the i-th restriction on the visiting order, where she must visit the shop numbered c_i after she visits the shop numbered d_i $(i = 1, \ldots, m)$.

There are no pair of j and k that satisfy $c_j = c_k$ and $d_j = d_k$.

Output

Output the minimum required walking length for her to move from the entrance to the exit. You should omit the length of her walk in the insides of shops.

Sample Input 1

```
10 3
3 7
8 9
2 5
```

Sample Output 1

```
23
```

Sample Input 2

```
10 3
8 9
6 7
2 4
```

Sample Output 2

```
19
```

Sample Input 3

```
10 0
```

Sample Output 3

```
11
```

Sample Input 4

```
10 6
6 7
4 5
2 5
6 9
3 5
6 8
```

Sample Output 4

```
23
```

Sample Input 5

```
1000 8
3 4
6 1000
5 1000
7 1000
8 1000
4 1000
9 1000
1 2
```

Sample Output 5

```
2997
```

Problem F.
Flipping Parentheses

Input: Standard Input
Time Limit: 5 seconds

A string consisting only of parentheses '(' and ')' is called balanced if it is one of the following.

- A string "()" is balanced.
- Concatenation of two balanced strings are balanced.
- When a string s is balanced, so is the concatenation of three strings "(", s, and ")" in this order.

Note that the condition is stronger than merely the numbers of '(' and ')' are equal. For instance, "())(()" is *not* balanced.

Your task is to keep a string in a balanced state, under a severe condition in which a cosmic ray may flip the direction of parentheses.

You are initially given a balanced string. Each time the direction of a single parenthesis is flipped, your program is notified the position of the changed character in the string. Then, calculate and output the *leftmost* position that, if the parenthesis there is flipped, the whole string gets back to the balanced state. After the string is balanced by changing the parenthesis indicated by your program, next cosmic ray flips another parenthesis, and the steps are repeated several times.

Input

The input consists of a single test case formatted as follows.

N Q
s
q_1
$\vdots$
q_Q

The first line consists of two integers N and Q ($2 \leq N \leq 300000$, $1 \leq Q \leq 150000$). The second line is a string s of balanced parentheses with length N. Each of the following Q lines is an integer q_i ($1 \leq q_i \leq N$) that indicates that the direction of the q_i-th parenthesis is flipped.

Output

For each event q_i, output the position of the leftmost parenthesis you need to flip in order to get back to the balanced state.

Note that each input flipping event q_i is applied to the string after the previous flip q_{i-1} and its fix.

Sample Input 1	Sample Output 1
6 3 ((())) 4 3 1	2 2 1

Sample Input 2	Sample Output 2
20 9 ()((((()))))()()()() 15 20 13 5 3 10 3 17 18	2 20 8 5 3 2 2 3 18

In the first sample, the initial state is "((()))". The 4th parenthesis is flipped and the string becomes "(((())". Then, to keep the balance you should flip the 2nd parenthesis and get "()(())". The next flip of the 3rd parenthesis is applied to the last state and yields "())())". To rebalance it, you have to change the 2nd parenthesis again yielding "(()())".

Problem G. Pair

At the interview senior *H&H* developers like to give newcomers different tasks. One of the tasks is the following. You are given the K different natural numbers from 1 to N. Your task is to find two of them so that one is divided by another.

Input

First line of the input contains numbers N and K ($2 \leq N \leq 600000, N/2 + 1 \leq K \leq N$). The second line contains K numbers.

Output

Write to the output indexes of the two numbers you found or two zeros if there are no such numbers.

Example

standard input	standard output
7 5 4 3 5 7 2	1 5

Problem H. HHPaint

The famous in the Volga region *H&H* company decided to create a very special graphic tool. The head of *H&H* believes that it can be used by scientists for numerical experiments. That's why one of the parts of the tool is "triangulating" module. And this module is one you are responsible for.

You are given N points. No three points lay on the same line. Lets consider the Convex polygon of minimal area containing all the points. Your task is to split the polygon to the set of not overlapped triangles in such a way, that all the vertices of the triangles are points from the given set. Each point should be used as a vertex of at least one triangle.

Input

The first line of the input file contains number of points N ($3 \le N \le 15000$). Each of the following N lines contains the coordinates of the corresponding point ($-10^6 \le x_i, y_i \le 10^6, 1 \le i \le N$). All coordinates are integers.

Output

Write in the first line of the output file the number of the triangles Q. Each of the following Q lines should contain description of the triangle. The description consists of three integers — numbers of points for each vertex of the triangle. The points are numbered from 1 to N in the order of their appearance in the input file. If there are several solutions, output any of them.

Example

standard input	standard output
6 -10 0 10 0 0 3 -1 4 1 4 0 5	7 1 2 3 1 3 4 1 4 6 2 3 5 2 5 6 6 4 5 3 4 5