Problem A. Counting Divisors

Time Limit 1000 ms
Mem Limit 524288 kB

Topicwise Problem Link

Given n integers, your task is to report for each integer the number of its divisors.

For example, if $x = 18$, the correct answer is 6 because its divisors are $1, 2, 3, 6, 9, 18$.

Input

The first input line has an integer n: the number of integers.

After this, there are n lines, each containing an integer x.

Output

For each integer, print the number of its divisors.

Constraints

- $1 \leq n \leq 10^5$
- $1 \leq x \leq 10^6$

Example

Input	Output
3 16 17 18	5 2 6

Problem B. Divide and Equalize

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

You are given an array a consisting of n positive integers. You can perform the following operation on it:

1. Choose a pair of elements a_i and a_j ($1 \le i, j \le n$ and $i \neq j$);
2. Choose one of the divisors of the integer a_i, i.e., an integer x such that $a_i \bmod x = 0$;
3. Replace a_i with $\frac{a_i}{x}$ and a_j with $a_j \cdot x$.

Determine whether it is possible to make all elements in the array the same by applying the operation a certain number of times (possibly zero).

For example, let's consider the array a = [100, 2, 50, 10, 1] with 5 elements. Perform two operations on it:

1. Choose $a_3 = 50$ and $a_2 = 2$, $x = 5$. Replace a_3 with $\frac{a_3}{x} = \frac{50}{5} = 10$, and a_2 with $a_2 \cdot x = 2 \cdot 5 = 10$. The resulting array is a = [100, 10, 10, 10, 1];
2. Choose $a_1 = 100$ and $a_5 = 1$, $x = 10$. Replace a_1 with $\frac{a_1}{x} = \frac{100}{10} = 10$, and a_5 with $a_5 \cdot x = 1 \cdot 10 = 10$. The resulting array is a = [10, 10, 10, 10, 10].

After performing these operations, all elements in the array a become equal to 10.

Input

The first line of the input contains a single integer t ($1 \le t \le 2000$) — the number of test cases.

Then follows the description of each test case.

The first line of each test case contains a single integer n ($1 \le n \le 10^4$) — the number of elements in the array a.

The second line of each test case contains exactly n integers a_i ($1 \le a_i \le 10^6$) — the elements of the array a.

It is guaranteed that the sum of n over all test cases does not exceed 10^4.

Output

For each test case, output a single line:

- "YES" if it is possible to make all elements in the array equal by applying the operation a certain (possibly zero) number of times;
- "NO" otherwise.

You can output the answer in any case (for example, the strings "yEs", "yes", "Yes", and "YES" will all be recognized as a positive answer).

Examples

Input	Output
7 5 100 2 50 10 1 3 1 1 1 4 8 2 4 2 4 30 50 27 20 2 75 40 2 4 4 3 2 3 1	YES YES NO YES NO YES NO

Note

The first test case is explained in the problem statement.

Problem C. T-primes

Time Limit	2000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

Topicwise Problem Link

We know that prime numbers are positive integers that have exactly two distinct positive divisors. Similarly, we'll call a positive integer t T-prime, if t has exactly three distinct positive divisors.

You are given an array of n positive integers. For each of them determine whether it is T-prime or not.

Input

The first line contains a single positive integer, n ($1 \le n \le 10^5$), showing how many numbers are in the array. The next line contains n space-separated integers x_i ($1 \le x_i \le 10^{12}$).

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is advised to use the `cin`, `cout` streams or the `%I64d` specifier.

Output

Print n lines: the i-th line should contain "`YES`" (without the quotes), if number x_i is T-prime, and "`NO`" (without the quotes), if it isn't.

Examples

Input	Output
3 4 5 6	YES NO NO

Note

The given test has three numbers. The first number 4 has exactly three divisors — 1, 2 and 4, thus the answer for this number is "YES". The second number 5 has two divisors (1 and 5), and the third number 6 has four divisors (1, 2, 3, 6), hence the answer for them is "NO".

Problem D. Composite Coloring

Time Limit 2000 ms
Mem Limit 524288 kB

Topicwise Problem Link

A positive integer is called *composite* if it can be represented as a product of two positive integers, both greater than 1. For example, the following numbers are composite: 6, 4, 120, 27. The following numbers aren't: 1, 2, 3, 17, 97.

Alice is given a sequence of n composite numbers $a_1, a_2, \ldots, a_n$.

She wants to choose an integer $m \le 11$ and color each element one of m colors from 1 to m so that:

- for each color from 1 to m there is at least one element of this color;
- each element is colored and colored exactly one color;
- the greatest common divisor of any two elements that are colored the same color is greater than 1, i.e. $\gcd(a_i, a_j) > 1$ for each pair i, j if these elements are colored the same color.

Note that equal elements can be colored different colors — you just have to choose one of m colors for each of the indices from 1 to n.

Alice showed already that if all $a_i \le 1000$ then she can always solve the task by choosing some $m \le 11$.

Help Alice to find the required coloring. Note that you don't have to minimize or maximize the number of colors, you just have to find the solution with some m from 1 to 11.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases. Then the descriptions of the test cases follow.

The first line of the test case contains a single integer n ($1 \le n \le 1000$) — the amount of numbers in a sequence a.

The second line of the test case contains n composite integers $a_1, a_2, \ldots, a_n$ ($4 \leq a_i \leq 1000$).

It is guaranteed that the sum of n over all test cases doesn't exceed 10^4.

Output

For each test case print 2 lines. The first line should contain a single integer m ($1 \leq m \leq 11$) — the number of used colors. Consider colors to be numbered from 1 to m. The second line should contain any coloring that satisfies the above conditions. Print n integers $c_1, c_2, \ldots, c_n$ ($1 \leq c_i \leq m$), where c_i is the color of the i-th element. If there are multiple solutions then you can print any of them. Note that you don't have to minimize or maximize the number of colors, you just have to find the solution with some m from 1 to 11.

Remember that each color from 1 to m should be used at least once. Any two elements of the same color should not be coprime (i.e. their GCD should be greater than 1).

Examples

Input	Output
3 3 6 10 15 2 4 9 23 437 519 865 808 909 391 194 291 237 395 323 365 511 497 781 737 871 559 731 697 779 841 961	1 1 1 1 2 2 1 11 4 7 8 10 7 3 10 7 7 8 3 1 1 5 5 9 2 2 3 3 4 11 6

Note

In the first test case, $\gcd(6, 10) = 2$, $\gcd(6, 15) = 3$ and $\gcd(10, 15) = 5$. Therefore, it's valid to color all elements the same color. Note that there are other colorings which satisfy Alice's requirement in this test case.

In the second test case there is only one element of each color, so the coloring definitely satisfies Alice's requirement.

Problem E. Interesting Ratio

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Recently, Misha at the IT Campus "NEIMARK" camp learned a new topic — the Euclidean algorithm.

He was somewhat surprised when he realized that $a \cdot b = lcm(a, b) \cdot gcd(a, b)$, where $gcd(a, b)$ — is the greatest common divisor (GCD) of the numbers a and b and $lcm(a, b)$ — is the least common multiple (LCM). Misha thought that since the product of LCM and GCD exists, it might be interesting to consider their quotient: $F(a, b) = \frac{lcm(a,b)}{gcd(a,b)}$.

For example, he took $a = 2$ and $b = 4$, computed $F(2, 4) = \frac{4}{2} = 2$ and obtained a prime number (a number is prime if it has exactly two divisors)! Now he considers $F(a, b)$ to be an interesting ratio if $a < b$ and $F(a, b)$ is a prime number.

Since Misha has just started studying number theory, he needs your help to calculate — how many different pairs of numbers a and b are there such that $F(a, b)$ is an interesting ratio and $1 \le a < b \le n$?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 10^3$). The description of the test cases follows.

A single line of each test case contains a single integer n ($2 \le n \le 10^7$).

It is guaranteed that the sum of n over all test cases does not exceed 10^7.

Output

For each test case, output the number of interesting ratios $F(a, b)$ for pairs satisfying $1 \le a < b \le n$.

Examples

Input	Output
4 5 10 34 10007	4 11 49 24317

Problem F. Almost Prime

Time Limit	2000 ms
Mem Limit	262144 kB
Input File	stdin
Output File	stdout

Topicwise Problem Link

A number is called almost prime if it has exactly two distinct prime divisors. For example, numbers 6, 18, 24 are almost prime, while 4, 8, 9, 42 are not. Find the amount of almost prime numbers which are between 1 and n, inclusive.

Input

Input contains one integer number n ($1 \leq n \leq 3000$).

Output

Output the amount of almost prime numbers between 1 and n, inclusive.

Examples

Input	Output
10	2

Input	Output
21	8

Problem G. Bachgold Problem

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Bachgold problem is very easy to formulate. Given a positive integer n represent it as a sum of **maximum possible** number of prime numbers. One can prove that such representation exists for any integer greater than 1.

Recall that integer k is called prime if it is greater than 1 and has exactly two positive integer divisors — 1 and k.

Input

The only line of the input contains a single integer n ($2 \le n \le 100\,000$).

Output

The first line of the output contains a single integer k — maximum possible number of primes in representation.

The second line should contain k primes with their sum equal to n. You can print them in any order. If there are several optimal solution, print any of them.

Examples

Input	Output
5	2 2 3

Input	Output
6	3 2 2 2

Problem H. Different Divisors

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Positive integer x is called *divisor* of positive integer y, if y is divisible by x without remainder. For example, 1 is a divisor of 7 and 3 is not divisor of 8.

We gave you an integer d and asked you to find **the smallest** positive integer a, such that

- a has at least 4 divisors;
- difference between any two divisors of a is at least d.

Input

The first line contains a single integer t ($1 \le t \le 3000$) — the number of test cases.

The first line of each test case contains a single integer d ($1 \le d \le 10000$).

Output

For each test case print one integer a — the answer for this test case.

Examples

Input	Output
2 1 2	6 15

Note

In the first test case, integer 6 have following divisors: $[1, 2, 3, 6]$. There are 4 of them and the difference between any two of them is at least 1. There is no smaller integer with at least 4 divisors.

In the second test case, integer 15 have following divisors: $[1, 3, 5, 15]$. There are 4 of them and the difference between any two of them is at least 2.

The answer 12 is INVALID because divisors are $[1, 2, 3, 4, 6, 12]$. And the difference between, for example, divisors 2 and 3 is less than $d = 2$.

Problem I. We Were Both Children

Time Limit 3000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Mihai and Slavic were looking at a group of n frogs, numbered from 1 to n, all initially located at point 0. Frog i has a hop length of a_i.

Each second, frog i hops a_i units forward. Before any frogs start hopping, Slavic and Mihai can place **exactly one** trap in a coordinate in order to catch all frogs that will ever pass through the corresponding coordinate.

However, the children can't go far away from their home so they can only place a trap in the first n points (that is, in a point with a coordinate between 1 and n) and the children can't place a trap in point 0 since they are scared of frogs.

Can you help Slavic and Mihai find out what is the maximum number of frogs they can catch using a trap?

Input

The first line of the input contains a single integer t ($1 \leq t \leq 100$) — the number of test cases. The description of test cases follows.

The first line of each test case contains a single integer n ($1 \leq n \leq 2 \cdot 10^5$) — the number of frogs, which equals the distance Slavic and Mihai can travel to place a trap.

The second line of each test case contains n integers $a_1, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) — the lengths of the hops of the corresponding frogs.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case output a single integer — the maximum number of frogs Slavic and Mihai can catch using a trap.

Examples

Input	Output
7 5 1 2 3 4 5 3 2 2 2 6 3 1 3 4 9 10 9 1 3 2 4 2 3 7 8 5 1 10 8 7 11 6 8 12 4 4 8 10 9 11 9 12 1 7 2 5 8 10	3 3 3 5 0 4 4

Note

In the first test case, the frogs will hop as follows:

- Frog 1: $0 \to 1 \to 2 \to 3 \to \mathbf{4} \to \cdots$
- Frog 2: $0 \to 2 \to \mathbf{4} \to 6 \to 8 \to \cdots$
- Frog 3: $0 \to 3 \to 6 \to 9 \to 12 \to \cdots$
- Frog 4: $0 \to \mathbf{4} \to 8 \to 12 \to 16 \to \cdots$
- Frog 5: $0 \to 5 \to 10 \to 15 \to 20 \to \cdots$

Therefore, if Slavic and Mihai put a trap at coordinate 4, they can catch three frogs: frogs 1, 2, and 4. It can be proven that they can't catch any more frogs.

In the second test case, Slavic and Mihai can put a trap at coordinate 2 and catch all three frogs instantly.

Problem J. Sherlock and his girlfriend

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Sherlock has a new girlfriend (so unlike him!). Valentine's day is coming and he wants to gift her some jewelry.

He bought n pieces of jewelry. The i-th piece has price equal to $i + 1$, that is, the prices of the jewelry are $2, 3, 4, \ldots n + 1$.

Watson gave Sherlock a challenge to color these jewelry pieces such that two pieces don't have the same color if the price of one piece is a prime divisor of the price of the other piece. Also, Watson asked him to minimize the number of different colors used.

Help Sherlock complete this trivial task.

Input

The only line contains single integer n ($1 \le n \le 100000$) — the number of jewelry pieces.

Output

The first line of output should contain a single integer k, the minimum number of colors that can be used to color the pieces of jewelry with the given constraints.

The next line should consist of n space-separated integers (between 1 and k) that specify the color of each piece in the order of increasing price.

If there are multiple ways to color the pieces using k colors, you can output any of them.

Examples

Input	Output
3	2 1 1 2

Input	Output
4	2 2 1 1 2

Note

In the first input, the colors for first, second and third pieces of jewelry having respective prices 2, 3 and 4 are 1, 1 and 2 respectively.

In this case, as 2 is a prime divisor of 4, colors of jewelry having prices 2 and 4 must be distinct.

Problem K. Bash's Big Day

Time Limit 2000 ms
Mem Limit 524288 kB

Topicwise Problem Link

Bash has set out on a journey to become the greatest Pokemon master. To get his first Pokemon, he went to Professor Zulu's Lab. Since Bash is Professor Zulu's favourite student, Zulu allows him to take as many Pokemon from his lab as he pleases.

But Zulu warns him that a group of $k > 1$ Pokemon with strengths $\{s_1, s_2, s_3, ..., s_k\}$ tend to fight among each other if $gcd(s_1, s_2, s_3, ..., s_k) = 1$ (see notes for gcd definition).

Bash, being smart, does not want his Pokemon to fight among each other. However, he also wants to maximize the number of Pokemon he takes from the lab. Can you help Bash find out the maximum number of Pokemon he can take?

Note: A Pokemon cannot fight with itself.

Input

The input consists of two lines.

The first line contains an integer n ($1 \le n \le 10^5$), the number of Pokemon in the lab.

The next line contains n space separated integers, where the i-th of them denotes s_i ($1 \le s_i \le 10^5$), the strength of the i-th Pokemon.

Output

Print single integer — the maximum number of Pokemons Bash can take.

Examples

Input	Output
3 2 3 4	2

Input	Output
5 2 3 4 6 7	3

Note

gcd (greatest common divisor) of positive integers set $\{a_1, a_2, ..., a_n\}$ is the maximum positive integer that divides all the integers $\{a_1, a_2, ..., a_n\}$.

In the first sample, we can take Pokemons with strengths $\{2, 4\}$ since $gcd(2, 4) = 2$.

In the second sample, we can take Pokemons with strengths $\{2, 4, 6\}$, and there is no larger group with $gcd \neq 1$.

Problem L. Insert and Equalize

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

You are given an integer array $a_1, a_2, \ldots, a_n$, all its elements are distinct.

First, you are asked to insert one more integer a_{n+1} into this array. a_{n+1} should not be equal to any of $a_1, a_2, \ldots, a_n$.

Then, you will have to make all elements of the array equal. At the start, you choose a **positive** integer x ($x > 0$). In one operation, you add x to exactly one element of the array. **Note that x is the same for all operations**.

What's the smallest number of operations it can take you to make all elements equal, after you choose a_{n+1} and x?

Input

The first line contains a single integer t ($1 \le t \le 10^4$) — the number of testcases.

The first line of each testcase contains a single integer n ($1 \le n \le 2 \cdot 10^5$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$). All a_i are distinct.

The sum of n over all testcases doesn't exceed $2 \cdot 10^5$.

Output

For each testcase, print a single integer — the smallest number of operations it can take you to make all elements equal, after you choose integers a_{n+1} and x.

Examples

Input	Output
3 3 1 2 3 5 1 -19 17 -3 -15 1 10	6 27 1

Note

In the first testcase, you can choose $a_{n+1} = 4$, the array becomes $[1, 2, 3, 4]$. Then choose $x = 1$ and apply the operation 3 times to the first element, 2 times to the second element, 1 time to the third element and 0 times to the fourth element.

In the second testcase, you can choose $a_{n+1} = 13, x = 4$.

In the third testcase, you can choose $a_{n+1} = 9, x = 1$. Then apply the operation once to a_{n+1}.

Problem M. Turtle Fingers: Count the Values of k

Time Limit 5000 ms
Mem Limit 262144 kB

Topicwise Problem Link

You are given three **positive** integers a, b and l ($a, b, l > 0$).

It can be shown that there always exists a way to choose **non-negative** (i.e. ≥ 0) integers k, x, and y such that $l = k \cdot a^x \cdot b^y$.

Your task is to find the number of distinct possible values of k across all such ways.

Input

The first line contains the integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The following t lines contain three integers, a, b and l ($2 \leq a, b \leq 100$, $1 \leq l \leq 10^6$) — description of a test case.

Output

Output t lines, with the i-th ($1 \leq i \leq t$) line containing an integer, the answer to the i-th test case.

Examples

Input	Output
11 2 5 20 2 5 21 4 6 48 2 3 72 3 5 75 2 2 1024 3 7 83349 100 100 1000000 7 3 2 2 6 6 17 3 632043	6 1 5 12 6 11 24 4 1 3 24

Note

In the first test case, $a = 2, b = 5, l = 20$. The possible values of k (and corresponding x, y) are as follows:

- Choose $k = 1, x = 2, y = 1$. Then $k \cdot a^x \cdot b^y = 1 \cdot 2^2 \cdot 5^1 = 20 = l$.
- Choose $k = 2, x = 1, y = 1$. Then $k \cdot a^x \cdot b^y = 2 \cdot 2^1 \cdot 5^1 = 20 = l$.
- Choose $k = 4, x = 0, y = 1$. Then $k \cdot a^x \cdot b^y = 4 \cdot 2^0 \cdot 5^1 = 20 = l$.
- Choose $k = 5, x = 2, y = 0$. Then $k \cdot a^x \cdot b^y = 5 \cdot 2^2 \cdot 5^0 = 20 = l$.
- Choose $k = 10, x = 1, y = 0$. Then $k \cdot a^x \cdot b^y = 10 \cdot 2^1 \cdot 5^0 = 20 = l$.
- Choose $k = 20, x = 0, y = 0$. Then $k \cdot a^x \cdot b^y = 20 \cdot 2^0 \cdot 5^0 = 20 = l$.

In the second test case, $a = 2, b = 5, l = 21$. Note that $l = 21$ is not divisible by either $a = 2$ or $b = 5$. Therefore, we can only set $x = 0, y = 0$, which corresponds to $k = 21$.

In the third test case, $a = 4, b = 6, l = 48$. The possible values of k (and corresponding x, y) are as follows:

- Choose $k = 2, x = 1, y = 1$. Then $k \cdot a^x \cdot b^y = 2 \cdot 4^1 \cdot 6^1 = 48 = l$.
- Choose $k = 3, x = 2, y = 0$. Then $k \cdot a^x \cdot b^y = 3 \cdot 4^2 \cdot 6^0 = 48 = l$.
- Choose $k = 8, x = 0, y = 1$. Then $k \cdot a^x \cdot b^y = 8 \cdot 4^0 \cdot 6^1 = 48 = l$.
- Choose $k = 12, x = 1, y = 0$. Then $k \cdot a^x \cdot b^y = 12 \cdot 4^1 \cdot 6^0 = 48 = l$.
- Choose $k = 48, x = 0, y = 0$. Then $k \cdot a^x \cdot b^y = 48 \cdot 4^0 \cdot 6^0 = 48 = l$.

Problem N. Turtle Tenacity: Continual Mods

Time Limit 2000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Given an array $a_1, a_2, \ldots, a_n$, determine whether it is possible to **rearrange its elements** into $b_1, b_2, \ldots, b_n$, such that $b_1 \bmod b_2 \bmod \ldots \bmod b_n \neq 0$.

Here $x \bmod y$ denotes the remainder from dividing x by y. Also, the modulo operations are calculated from left to right. That is, $x \bmod y \bmod z = (x \bmod y) \bmod z$. For example, $2024 \bmod 1000 \bmod 8 = (2024 \bmod 1000) \bmod 8 = 24 \bmod 8 = 0$.

Input

The first line of the input contains a single integer t ($1 \leq t \leq 10^4$) — the number of test cases.

The first line of each test case contains a single integer n ($2 \leq n \leq 10^5$).

The second line of each test case contains n integers $a_1, a_2, \ldots, a_n$ ($1 \leq a_i \leq 10^9$).

The sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, output "`YES`" if it is possible, "`NO`" otherwise.

You can output the answer in any case (upper or lower). For example, the strings "`yEs`", "`yes`", "`Yes`", and "`YES`" will be recognized as positive responses.

Examples

Input	Output
8 6 1 2 3 4 5 6 5 3 3 3 3 3 3 2 2 3 5 1 1 2 3 7 3 1 2 2 3 1 1 2 6 5 2 10 10 10 2 4 3 6 9 3	YES NO YES NO YES NO YES NO

Note

In the first test case, rearranging the array into $b = [1, 2, 3, 4, 5, 6]$ (doing nothing) would result in $1 \bmod 2 \bmod 3 \bmod 4 \bmod 5 \bmod 6 = 1$. Hence it is possible to achieve the goal.

In the second test case, the array b must be equal to $[3, 3, 3, 3, 3]$, which would result in $3 \bmod 3 \bmod 3 \bmod 3 \bmod 3 = 0$. Hence it is impossible to achieve the goal.

In the third test case, rearranging the array into $b = [3, 2, 2]$ would result in $3 \bmod 2 \bmod 2 = 1$. Hence it is possible to achieve the goal.

Problem O. Collatz Conjecture

Time Limit 1000 ms
Mem Limit 262144 kB

Topicwise Problem Link

Recently, the first-year student Maxim learned about the Collatz conjecture, but he didn't pay much attention during the lecture, so he believes that the following process is mentioned in the conjecture:

There is a variable x and a constant y. The following operation is performed k times:

- increase x by 1, then
- while the number x is divisible by y, divide it by y.

Note that both of these actions are performed sequentially within one operation.

For example, if the number $x = 16$, $y = 3$, and $k = 2$, then after one operation x becomes 17, and after another operation x becomes 2, because after adding one, $x = 18$ is divisible by 3 twice.

Given the initial values of x, y, and k, Maxim wants to know what is the final value of x.

Input

Each test consists of multiple test cases. The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases. Then follows the description of the test cases.

The only line of each test case contains three integers x, y, and k ($1 \le x, k \le 10^9$, $2 \le y \le 10^9$) — the initial variable, constant and the number of operations.

Output

For each test case, output a single integer — the number obtained after applying k operations.

Examples

Input	Output
13 1 3 1 2 3 1 24 5 5 16 3 2 2 2 1 1337 18 1 1 2 144133 12345678 3 10 998244353 2 998244353 998244353 123456789 998244352 998244354 998241111 998244352 998244355 2 9982443 1000000000 1000000000 1000000000	2 1 1 2 3 1338 1 16936 1 21180097 6486 1 2

Note

In the first test case, there is only one operation applied to $x = 1$, resulting in x becoming 2.

In the second test case, for $x = 2$, within one operation, one is added to x and it's divided by $y = 3$, resulting in x becoming 1.

In the third test case, x changes as follows:

- After the first operation, $x = 1$, because $24 + 1 = 25$ and 25 is divisible by $y = 5$ twice within one operation.
- After the second operation, $x = 2$.
- After the third operation, $x = 3$.
- After the fourth operation, $x = 4$.
- After the fifth operation, $x = 1$.