Problem A. Bachgold Problem

Time limit 1000 ms
Mem limit 262144 kB
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 ($2 \leq n \leq 100\,000$).

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 B. Relatively Prime Pairs

Time limit 2000 ms
Mem limit 262144 kB

[Problem Link]

You are given a set of all integers from l to r inclusive, $l < r$, $(r - l + 1) \leq 3 \cdot 10^5$ and $(r - l)$ is always odd.

You want to split these numbers into exactly $\frac{r-l+1}{2}$ pairs in such a way that for each pair (i, j) the greatest common divisor of i and j is equal to 1. Each number should appear in exactly one of the pairs.

Print the resulting pairs or output that no solution exists. If there are multiple solutions, print any of them.

Input

The only line contains two integers l and r ($1 \leq l < r \leq 10^{18}$, $r - l + 1 \leq 3 \cdot 10^5$, $(r - l)$ is odd).

Output

If any solution exists, print "`YES`" in the first line. Each of the next $\frac{r-l+1}{2}$ lines should contain some pair of integers. *GCD* of numbers in each pair should be equal to 1. All $(r - l + 1)$ numbers should be pairwise distinct and should have values from l to r inclusive.

If there are multiple solutions, print any of them.

If there exists no solution, print "`NO`".

Examples

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td>1 8</td><td>YES
2 7
4 1
3 8
6 5</td></tr>
</table>

Problem C. T-primes

Time limit	2000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

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. k-th divisor

Time limit 2000 ms
Mem limit 262144 kB
Problem Link

You are given two integers n and k. Find k-th smallest divisor of n, or report that it doesn't exist.

Divisor of n is any such natural number, that n can be divided by it without remainder.

Input

The first line contains two integers n and k ($1 \le n \le 10^{15}$, $1 \le k \le 10^9$).

Output

If n has less than k divisors, output `-1`.

Otherwise, output the k-th smallest divisor of n.

Examples

Input	Output
4 2	2

Input	Output
5 3	-1

Input	Output
12 5	6

Note

In the first example, number 4 has three divisors: 1, 2 and 4. The second one is 2.

In the second example, number 5 has only two divisors: 1 and 5. The third divisor doesn't exist, so the answer is `-1`.

Problem E. Fadi and LCM

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

Today, Osama gave Fadi an integer X, and Fadi was wondering about the minimum possible value of $max(a, b)$ such that $LCM(a, b)$ equals X. Both a and b should be positive integers.

$LCM(a, b)$ is the smallest positive integer that is divisible by both a and b. For example, $LCM(6, 8) = 24$, $LCM(4, 12) = 12$, $LCM(2, 3) = 6$.

Of course, Fadi immediately knew the answer. Can you be just like Fadi and find any such pair?

Input

The first and only line contains an integer X ($1 \leq X \leq 10^{12}$).

Output

Print two positive integers, a and b, such that the value of $max(a, b)$ is minimum possible and $LCM(a, b)$ equals X. If there are several possible such pairs, you can print any.

Examples

Input	Output
2	1 2

Input	Output
6	2 3

Input	Output
4	1 4

Input	Output
1	1 1

Problem F. Maximum GCD

Time limit 1000 ms
Mem limit 262144 kB
Problem Link

Let's consider all integers in the range from 1 to n (inclusive).

Among all pairs of **distinct** integers in this range, find the maximum possible greatest common divisor of integers in pair. Formally, find the maximum value of $\gcd(a, b)$, where $1 \le a < b \le n$.

The greatest common divisor, $\gcd(a, b)$, of two positive integers a and b is the biggest integer that is a divisor of both a and b.

Input

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

The only line of each test case contains a single integer n ($2 \le n \le 10^6$).

Output

For each test case, output the maximum value of $\gcd(a, b)$ among all $1 \le a < b \le n$.

Examples

Input	Output
2 3 5	1 2

Note

In the first test case, $\gcd(1, 2) = \gcd(2, 3) = \gcd(1, 3) = 1$.

In the second test case, 2 is the maximum possible value, corresponding to $\gcd(2, 4)$.

Problem G. Bash's Big Day

Time limit 2000 ms
Mem limit 524288 kB
[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 ($1 \leq n \leq 10^5$), the number of Pokemon in the lab.

The next line contains n space separated integers, where the i-th of them denotes s_i ($1 \leq s_i \leq 10^5$), 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 $\{a_1, a_2, \ldots, a_n\}$ is the maximum positive integer that divides all the integers $\{a_1, a_2, \ldots, a_n\}$.

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 H. Taxes

Time limit	2000 ms
Mem limit	262144 kB
Input file	stdin
Output file	stdout

Problem Link

Mr. Funt now lives in a country with a very specific tax laws. The total income of mr. Funt during this year is equal to n ($n \geq 2$) burles and the amount of tax he has to pay is calculated as the maximum divisor of n (not equal to n, of course). For example, if $n = 6$ then Funt has to pay 3 burles, while for $n = 25$ he needs to pay 5 and if $n = 2$ he pays only 1 burle.

As mr. Funt is a very opportunistic person he wants to cheat a bit. In particular, he wants to split the initial n in several parts $n_1 + n_2 + \ldots + n_k = n$ (here k is arbitrary, even $k = 1$ is allowed) and pay the taxes for each part separately. He can't make some part equal to 1 because it will reveal him. So, the condition $n_i \geq 2$ should hold for all i from 1 to k.

Ostap Bender wonders, how many money Funt has to pay (i.e. minimal) if he chooses and optimal way to split n in parts.

Input

The first line of the input contains a single integer n ($2 \leq n \leq 2 \cdot 10^9$) — the total year income of mr. Funt.

Output

Print one integer — minimum possible number of burles that mr. Funt has to pay as a tax.

Examples

Input	Output
4	2

Input	Output
27	3

Problem I. Di-visible Confusion

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

YouKn0wWho has an integer sequence $a_1, a_2, \ldots, a_n$. He will perform the following operation until the sequence becomes empty: select an index i such that $1 \le i \le |a|$ and a_i is **not** divisible by $(i+1)$, and erase this element from the sequence. Here $|a|$ is the length of sequence a at the moment of operation. Note that the sequence a changes and the next operation is performed on this changed sequence.

For example, if $a = [3, 5, 4, 5]$, then he can select $i = 2$, because $a_2 = 5$ is not divisible by $i + 1 = 3$. After this operation the sequence is $[3, 4, 5]$.

Help YouKn0wWho determine if it is possible to erase the whole sequence using the aforementioned operation.

Input

The first line contains a single integer t $(1 \le t \le 10\,000)$ — the number of test cases.

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

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

It is guaranteed that the sum of n over all test cases doesn't exceed $3 \cdot 10^5$.

Output

For each test case, print "`YES`" (without quotes) if it is possible to erase the whole sequence using the aforementioned operation, print "`NO`" (without quotes) otherwise. You can print each letter in any register (upper or lower).

Examples

Input	Output
5 3 1 2 3 1 2 2 7 7 10 384836991 191890310 576823355 782177068 40 8 6 69 696 69696 696969 6969696 69696969 696	YES NO YES YES NO

Note

In the first test case, YouKn0wWho can perform the following operations (the erased elements are underlined): $[1, \underline{2}, 3] \to [\underline{1}, 3] \to [\underline{3}] \to [\,]$.

In the second test case, it is impossible to erase the sequence as i can only be 1, and when $i = 1$, $a_1 = 2$ is divisible by $i + 1 = 2$.

Problem J. Joty and Chocolate

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

Little Joty has got a task to do. She has a line of n tiles indexed from 1 to n. She has to paint them in a strange pattern.

An unpainted tile should be painted Red if it's index is divisible by a and an unpainted tile should be painted Blue if it's index is divisible by b. So the tile with the number divisible by a and b can be either painted Red or Blue.

After her painting is done, she will get p chocolates for each tile that is painted Red and q chocolates for each tile that is painted Blue.

Note that she can paint tiles in any order she wants.

Given the required information, find the maximum number of chocolates Joty can get.

Input

The only line contains five integers n, a, b, p and q ($1 \leq n, a, b, p, q \leq 10^9$).

Output

Print the only integer s — the maximum number of chocolates Joty can get.

Note that the answer can be too large, so you should use 64-bit integer type to store it. In `C++` you can use the `long long` integer type and in `Java` you can use `long` integer type.

Examples

Input	Output
5 2 3 12 15	39

Input	Output
20 2 3 3 5	51

Problem K. Swords

Time limit 2000 ms
Mem limit 262144 kB

Problem Link

There were n types of swords in the theater basement which had been used during the plays. Moreover there were **exactly** x swords of each type. y people have broken into the theater basement and each of them has taken exactly z swords of some **single type**. Note that different people might have taken different types of swords. Note that the values x, y and z are unknown for you.

The next morning the director of the theater discovers the loss. He counts all swords — exactly a_i swords of the i-th type are left untouched.

The director has no clue about the initial number of swords of each type in the basement, the number of people who have broken into the basement and how many swords each of them have taken.

For example, if $n = 3, a = [3, 12, 6]$ then one of the possible situations is $x = 12, y = 5$ and $z = 3$. Then the first three people took swords of the first type and the other two people took swords of the third type. Note that you don't know values x, y and z beforehand but know values of n and a.

Thus he seeks for your help. Determine the **minimum** number of people y, which could have broken into the theater basement, and the number of swords z each of them has taken.

Input

The first line of the input contains one integer n $(2 \le n \le 2 \cdot 10^5)$ — the number of types of swords.

The second line of the input contains the sequence $a_1, a_2, \dots, a_n$ $(0 \le a_i \le 10^9)$, where a_i equals to the number of swords of the i-th type, which have remained in the basement after the theft. It is guaranteed that there exists at least one such pair of indices (j, k) that $a_j \ne a_k$.

Output

Print two integers y and z — the minimum number of people which could have broken into the basement and the number of swords each of them has taken.

Examples

Input	Output
3 3 12 6	5 3

Input	Output
2 2 9	1 7

Input	Output
7 2 1000000000 4 6 8 4 2	2999999987 2

Input	Output
6 13 52 0 13 26 52	12 13

Note

In the first example the minimum value of y equals to 5, i.e. the minimum number of people who could have broken into the basement, is 5. Each of them has taken 3 swords: three of them have taken 3 swords of the first type, and two others have taken 3 swords of the third type.

In the second example the minimum value of y is 1, i.e. the minimum number of people who could have broken into the basement, equals to 1. He has taken 7 swords of the first type.

Problem L. High School: Become Human

Time limit 1000 ms
Mem limit 262144 kB
Problem Link

Year 2118. Androids are in mass production for decades now, and they do all the work for humans. But androids have to go to school to be able to solve creative tasks. Just like humans before.

It turns out that high school struggles are not gone. If someone is not like others, he is bullied. Vasya-8800 is an economy-class android which is produced by a little-known company. His design is not perfect, his characteristics also could be better. So he is bullied by other androids.

One of the popular pranks on Vasya is to force him to compare x^y with y^x. Other androids can do it in milliseconds while Vasya's memory is too small to store such big numbers.

Please help Vasya! Write a fast program to compare x^y with y^x for Vasya, maybe then other androids will respect him.

Input

On the only line of input there are two integers x and y ($1 \leq x, y \leq 10^9$).

Output

If $x^y < y^x$, then print '<' (without quotes). If $x^y > y^x$, then print '>' (without quotes). If $x^y = y^x$, then print '=' (without quotes).

Examples

Input	Output
5 8	>

Input	Output
`10 3`	`<`

Input	Output
`6 6`	`=`

Note

In the first example $5^8 = 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 \cdot 5 = 390625$, and $8^5 = 8 \cdot 8 \cdot 8 \cdot 8 \cdot 8 = 32768$. So you should print '>'.

In the second example $10^3 = 1000 < 3^{10} = 59049$.

In the third example $6^6 = 46656 = 6^6$.

Problem M. Turtle Fingers: Count the Values of k

Time limit 5000 ms
Mem limit 262144 kB

[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. Composite Coloring

Time limit 2000 ms
Mem limit 524288 kB
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 $m \leq 11$ 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 $a_i \leq 1000$ then she can always solve the task by choosing some $m \leq 11$.

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 ($1 \leq t \leq 1000$) — 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 ($1 \leq n \leq 1000$) — 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 ($1 \le m \le 11$) — 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 $c_1, c_2, \dots, c_n$ ($1 \le c_i \le m$), 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 32	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

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 O. Different Divisors

Time limit 1000 ms
Mem limit 262144 kB

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 $(1 \leq t \leq 3000)$ — the number of test cases.

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

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 P. Divide and Equalize

Time limit	2000 ms
Mem limit	262144 kB

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 $i \ne j$);
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 Q. Paint the Array

Time limit 2000 ms
Mem limit 262144 kB
Problem Link

You are given an array a consisting of n positive integers. You have to choose a positive integer d and paint all elements into two colors. All elements which are divisible by d will be painted red, and all other elements will be painted blue.

The coloring is called beautiful if there are no pairs of adjacent elements with the same color in the array. Your task is to find any value of d which yields a beautiful coloring, or report that it is impossible.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of testcases.

The first line of each testcase contains one integer n ($2 \le n \le 100$) — the number of elements of the array.

The second line of each testcase contains n integers $a_1, a_2, \dots, a_n$ ($1 \le a_i \le 10^{18}$).

Output

For each testcase print a single integer. If there is no such value of d that yields a beautiful coloring, print 0. Otherwise, print any suitable value of d ($1 \le d \le 10^{18}$).

Examples

Input	Output
5 5 1 2 3 4 5 3 10 5 15 3 100 10 200 10 9 8 2 6 6 2 8 6 5 4 2 1 3	2 0 100 0 3

Problem R. Chef and Prime Divisors

Time limit	1000 ms
Code length Limit	50000 B
OS	Linux

Read problems statements in Mandarin Chinese and Russian.

You are given two positive integers – **A** and **B**. You have to check whether **A** is divisible by all the prime divisors of **B**.

Input

The first line of the input contains an integer **T** denoting the number of test cases. The description of **T** test cases follows.

For each test case, you are given two space separated integers – **A** and **B**.

Output

For each test case, output "Yes" (without quotes) if **A** contains all prime divisors of **B**, otherwise print "No".

Constraints

- $1 \le T \le 10^4$
- $1 \le A, B \le 10^{18}$

Subtasks

- **Subtask 1 (20 points):** $1 \le B \le 10^7$
- **Subtask 2 (30 points):** $1 \le A \le 10^7$
- **Subtask 3 (50 points):** Original constraints

Problem Link

Sample 1

Input	Output
3 120 75 128 16 7 8	Yes Yes No

Example case 1. In the first case 120 = 2^3*3*5 and 75 = 3*5^2. 120 is divisible by both 3 and 5. Hence, we will print "Yes"

Example case 2. In the second case both 128 and 16 are powers of two. Hence, the answer is "Yes"

Example case 3. In the third case 8 is power of two and 7 is not divisible by 2. So, the answer is "No"

Problem S. Non-coprime Split

Time limit 1000 ms
Mem limit 262144 kB
Problem Link

You are given two integers $l \le r$. You need to find **positive** integers a and b such that the following conditions are simultaneously satisfied:

- $l \le a + b \le r$
- $\gcd(a, b) \neq 1$

or report that they do not exist.

$\gcd(a, b)$ denotes the greatest common divisor of numbers a and b. For example, $\gcd(6, 9) = 3$, $\gcd(8, 9) = 1$, $\gcd(4, 2) = 2$.

Input

The first line of the input contains an integer t ($1 \le t \le 500$) — the number of test cases.

Then the descriptions of the test cases follow.

The only line of the description of each test case contains 2 integers l, r ($1 \le l \le r \le 10^7$).

Output

For each test case, output the integers a, b that satisfy all the conditions on a separate line. If there is no answer, instead output a single number -1.

If there are multiple answers, you can output any of them.

Examples

Input	Output
11 11 15 1 3 18 19 41 43 777 777 8000000 10000000 2000 2023 1791791 1791791 1 4 2 3 9840769 9840769	6 9 -1 14 4 36 6 111 666 4000000 5000000 2009 7 -1 2 2 -1 6274 9834495

Note

In the first test case, $11 \le 6 + 9 \le 15$, $\gcd(6, 9) = 3$, and all conditions are satisfied. Note that this is not the only possible answer, for example, $\{4, 10\}$, $\{5, 10\}$, $\{6, 6\}$ are also valid answers for this test case.

In the second test case, the only pairs $\{a, b\}$ that satisfy the condition $1 \le a + b \le 3$ are $\{1, 1\}$, $\{1, 2\}$, $\{2, 1\}$, but in each of these pairs $\gcd(a, b)$ equals 1, so there is no answer.

In the third sample test, $\gcd(14, 4) = 2$.

Problem T. No Prime Differences

Time limit 2000 ms
Mem limit 262144 kB
Problem Link

You are given integers n and m. Fill an n by m grid with the integers 1 through $n \cdot m$, in such a way that for any two adjacent cells in the grid, the absolute difference of the values in those cells is not a prime number. Two cells in the grid are considered adjacent if they share a side.

16	7	1	9
12	8	2	3
13	4	10	11
14	5	6	15

It can be shown that under the given constraints, there is always a solution.

Input

The first line of the input contains a single integer t ($1 \le t \le 1000$) — the number of test cases. The description of the test cases follows.

The first and only line of each test case contains two integers n and m ($4 \le n, m \le 1000$) — the dimensions of the grid.

It is guaranteed that the sum of $n \cdot m$ over all test cases does not exceed 10^6.

Output

For each test case, output n lines of m integers each, representing the final grid. Every number from 1 to $n \cdot m$ should appear exactly once in the grid.

The extra spaces and blank lines in the sample output below are only present to make the output easier to read, and are **not** required.

If there are multiple solutions, print any of them.

Examples

Input	Output
3 4 4 5 7 6 4	16 7 1 9 12 8 2 3 13 4 10 11 14 5 6 15 29 23 17 9 5 6 2 33 27 21 15 11 7 1 32 31 25 19 20 16 10 26 30 24 18 14 8 4 35 34 28 22 13 12 3 2 3 7 11 8 9 1 10 17 13 5 4 18 14 6 12 19 23 15 21 20 24 16 22

Note

The first sample case corresponds to the picture above. The only absolute differences between adjacent elements in this grid are $1, 4, 6, 8$, and 9, none of which are prime.

Problem U. GCD Partition

Time limit 1000 ms
Mem limit 262144 kB
Problem Link

While at Kira's house, Josuke saw a piece of paper on the table with a task written on it.

The task sounded as follows. There is an array a of length n. On this array, do the following:

- select an integer $k > 1$;
- split the array into k subsegments [†];
- calculate the sum in each of k subsegments and write these sums to another array b (where the sum of the subsegment (l, r) is $\sum_{j=l}^{r} a_j$);
- the final score of such a split will be $\gcd(b_1, b_2, \ldots, b_k)$ [‡].

The task is to find such a partition that the score is **maximum possible**. Josuke is interested in this task but is not strong in computer science. Help him to find the maximum possible score.

[†] A division of an array into k subsegments is k pairs of numbers $(l_1, r_1), (l_2, r_2), \ldots, (l_k, r_k)$ such that $l_i \leq r_i$ and for every $1 \leq j \leq k-1$ $l_{j+1} = r_j + 1$, also $l_1 = 1$ and $r_k = n$. These pairs represent the subsegments.

[‡] $\gcd(b_1, b_2, \ldots, b_k)$ stands for the greatest common divisor (GCD) of the array b.

Input

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

For each test case, the first line contains one integer n ($2 \leq n \leq 2 \cdot 10^5$) — the length of the array a.

The second line contains n integers $a_1, a_2, a_3, \ldots, a_n$ ($1 \leq a_i \leq 10^9$) — the array a itself.

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

Output

For each test case print a single integer — the maximum score for the optimal partition.

Examples

Input	Output
6 4 2 2 1 3 2 1 2 3 1 4 5 6 1 2 1 1 1 3 10 12 30 37 88 12 78 89 17 2 12 6 7 7 7 7 7 7	4 1 5 3 1 21

Note

In the first test case, you can choose $k = 2$ and split the array into subsegments $(1, 2)$ and $(3, 4)$.

Then the score of such a partition will be equal to $\gcd(a_1 + a_2, a_3 + a_4) = \gcd(2 + 2, 1 + 3) = \gcd(4, 4) = 4$.

In the fourth test case, you can choose $k = 3$ and split the array into subsegments $(1, 2), (3, 5), (6, 6)$.

The split score is $\gcd(1 + 2, 1 + 1 + 1, 3) = 3$.

Problem V. Nastia and Nearly Good Numbers

Time limit 1000 ms
Mem limit 262144 kB

Problem Link

Nastia has 2 positive integers A and B. She defines that:

- The integer is good if it is divisible by $A \cdot B$;
- Otherwise, the integer is nearly good, if it is divisible by A.

For example, if $A = 6$ and $B = 4$, the integers 24 and 72 are good, the integers 6, 660 and 12 are nearly good, the integers 16, 7 are neither good nor nearly good.

Find 3 **different** positive integers x, y, and z such that **exactly one** of them is good and the **other** 2 are nearly good, and $x + y = z$.

Input

The first line contains a single integer t ($1 \le t \le 10\,000$) — the number of test cases.

The first line of each test case contains two integers A and B ($1 \le A \le 10^6$, $1 \le B \le 10^6$) — numbers that Nastia has.

Output

For each test case print:

- "`YES`" and 3 **different** positive integers x, y, and z ($1 \le x, y, z \le 10^{18}$) such that **exactly one** of them is good and the **other** 2 are nearly good, and $x + y = z$.
- "`NO`" if no answer exists.

You can print each character of "`YES`" or "`NO`" in any case.

If there are multiple answers, print any.

Examples

Input	Output
3 5 3 13 2 7 11	YES 10 50 60 YES 169 39 208 YES 28 154 182

Note

In the first test case: 60 — good number; 10 and 50 — nearly good numbers.

In the second test case: 208 — good number; 169 and 39 — nearly good numbers.

In the third test case: 154 — good number; 28 and 182 — nearly good numbers.

Problem W. Coprime

Time limit 3000 ms
Mem limit 262144 kB

Problem Link

Given an array of n positive integers $a_1, a_2, \ldots, a_n$ ($1 \le a_i \le 1000$). Find the maximum value of $i + j$ such that a_i and a_j are coprime,[†] or -1 if no such i, j exist.

For example consider the array $[1, 3, 5, 2, 4, 7, 7]$. The maximum value of $i + j$ that can be obtained is $5 + 7$, since $a_5 = 4$ and $a_7 = 7$ are coprime.

[†] Two integers p and q are coprime if the only positive integer that is a divisor of both of them is 1 (that is, their greatest common divisor is 1).

Input

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

The first line of each test case contains an integer n ($2 \le n \le 2 \cdot 10^5$) — the length of the array.

The following line contains n space-separated positive integers a_1, a_2,..., a_n ($1 \le a_i \le 1000$) — the elements of the array.

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 value of $i + j$ such that i and j satisfy the condition that a_i and a_j are coprime, or output -1 in case no i, j satisfy the condition.

Examples

Input	Output
6 3 3 2 1 7 1 3 5 2 4 7 7 5 1 2 3 4 5 3 2 2 4 6 5 4 3 15 12 16 5 1 2 2 3 6	6 12 9 -1 10 7

Note

For the first test case, we can choose $i = j = 3$, with sum of indices equal to 6, since 1 and 1 are coprime.

For the second test case, we can choose $i = 7$ and $j = 5$, with sum of indices equal to $7 + 5 = 12$, since 7 and 4 are coprime.