Problem A. Rudolf and the Another Competition

Time limit 1000 ms
Mem limit 262144 kB

Rudolf has registered for a programming competition that will follow the rules of ICPC. The rules imply that for each solved problem, a participant gets 1 point, and also incurs a penalty equal to the number of minutes passed from the beginning of the competition to the moment of solving the problem. In the final table, the participant with the most points is ranked higher, and in case of a tie in points, the participant with the lower penalty is ranked higher.

In total, n participants have registered for the competition. Rudolf is a participant with index 1. It is known that m problems will be proposed. And the competition will last h minutes.

A powerful artificial intelligence has predicted the values $t_{i,j}$, which represent the number of minutes it will take for the i-th participant to solve the j-th problem.

Rudolf realized that the order of solving problems will affect the final result. For example, if $h = 120$, and the times to solve problems are $[20, 15, 110]$, then if Rudolf solves the problems in the order:

- $3, 1, 2$, then he will only solve the third problem and get 1 point and 110 penalty.
- $1, 2, 3$, then he will solve the first problem after 20 minutes from the start, the second one after $20 + 15 = 35$ minutes, and he will not have time to solve the third one. Thus, he will get 2 points and $20 + 35 = 55$ penalty.
- $2, 1, 3$, then he will solve the second problem after 15 minutes from the start, the first one after $15 + 20 = 35$ minutes, and he will not have time to solve the third one. Thus, he will get 2 points and $15 + 35 = 50$ penalty.

Rudolf became interested in what place he will take in the competition if each participant solves problems in the optimal order based on the predictions of the artificial intelligence. It will be assumed that in case of a tie in points and penalty, Rudolf will take the best place.

Input

The first line contains an integer t ($1 \leq t \leq 10^3$) — the number of test cases.

Then follow the descriptions of the test cases.

The first line of each test case contains three integers n, m, h ($1 \le n \cdot m \le 2 \cdot 10^5, 1 \le h \le 10^6$) — the number of participants, the number of problems, and the duration of the competition, respectively.

Then there are n lines, each containing m integers $t_{i,j}$ ($1 \le t_{i,j} \le 10^6$) — the number of minutes it will take for the i-th participant to solve the j-th problem.

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

Output

For each test case, output an integer — Rudolf's place in the final table if all participants solve problems in the optimal order.

Examples

<table>
<tr><th>Input</th><th>Output</th></tr>
<tr><td><pre>5
3 3 120
20 15 110
90 90 100
40 40 40
2 1 120
30
30
1 3 120
10 20 30
3 2 27
8 9
10 7
10 8
3 3 15
7 2 6
7 5 4
1 9 8</pre></td><td><pre>2
1
1
2
1</pre></td></tr>
</table>

Note

In the first example, Rudolf will get 2 points and 50 penalty minutes. The second participant will solve only one problem and get 1 point and 90 penalty minutes. And the third participant will solve all 3 problems and get 3 points and 240 penalty minutes. Thus, Rudolf will take the second place.

In the second example, both participants will get 1 point and 30 penalty minutes. In case of a tie in points, Rudolf gets the better position, so he will take the first place.

In the third example, Rudolf is the only participant, so he will take the first place.

In the fourth example, all participants can solve two problems with penalty of $25 = 8 + (8 + 9)$, $24 = 7 + (7 + 10)$ and $26 = 8 + (8 + 10)$, respectively, thanks to the penalty, the second participant gets the first place, and Rudolf gets the second.

Problem B. Minimum LCM

Time limit 2000 ms
Mem limit 524288 kB

You are given an integer n.

Your task is to find two positive (greater than 0) integers a and b such that $a + b = n$ and the least common multiple (LCM) of a and b is the minimum among all possible values of a and b. If there are multiple answers, you can print any of them.

Input

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

The first line of each test case contains a single integer n ($2 \leq n \leq 10^9$).

Output

For each test case, print two positive integers a and b — the answer to the problem. If there are multiple answers, you can print any of them.

Examples

Input	Output
4 2 9 5 10	1 1 3 6 1 4 5 5

Note

In the second example, there are 8 possible pairs of a and b:

- $a = 1, b = 8, LCM(1, 8) = 8;$
- $a = 2, b = 7, LCM(2, 7) = 14;$
- $a = 3, b = 6, LCM(3, 6) = 6;$
- $a = 4, b = 5, LCM(4, 5) = 20;$

- $a = 5, b = 4, LCM(5, 4) = 20;$
- $a = 6, b = 3, LCM(6, 3) = 6;$
- $a = 7, b = 2, LCM(7, 2) = 14;$
- $a = 8, b = 1, LCM(8, 1) = 8.$

In the third example, there are 5 possible pairs of a and b:

- $a = 1, b = 4, LCM(1, 4) = 4;$
- $a = 2, b = 3, LCM(2, 3) = 6;$
- $a = 3, b = 2, LCM(3, 2) = 6;$
- $a = 4, b = 1, LCM(4, 1) = 4.$

Problem C. Odd One Out

Time limit	1000 ms
Mem limit	262144 kB
OS	Windows

Given an integer X, let $f(X)$ denote the number of **ordered pairs** (a, b) such that $\gcd(a, b) \times \operatorname{lcm}(a, b) = X$.

A number Y is beautiful if $f(Y)$ is **odd**.

Count the number of beautiful numbers that lie in a given range $[L, R]$.

Refer to the notes below if you don't understand some of these terms.

Input

Input starts with an integer T ($1 \leq T \leq 10$), denoting the number of test cases.

Each case starts with a line containing two integers L and R ($1 \leq L \leq R \leq 10^9$).

Output

For each query, you have to print a line containing the number of beautiful numbers in the range $[L, R]$.

Examples

Input	Output
2 2 8 4669 176420	1 352

Note

The first three beautiful numbers are $1, 4, 9$.

In the first sample, the only beautiful number which lies between 2 and 8 is the number 4. It is a beautiful number because there are 3 ordered pairs (a, b) satisfying the equation given in the statement: they are $(1, 4)$, $(2, 2)$, $(4, 1)$. Since this number 3 is odd, the number 4 qualifies as beautiful. One can check that the other numbers between 2 and 8 do not satisfy these conditions.

Definitions

- An ordered pair (a, b) is a pair where the order in which the elements a, b appear is significant. The ordered pair (a, b) is different from the ordered pair (b, a) unless $a = b$.
- The greatest common divisor $\gcd(a, b)$ of two numbers a, b is the largest number which divides both a and b. For example: $\gcd(6, 15) = 3$.
- The least common multiply $\operatorname{lcm}(a, b)$ of two numbers a, b is the smallest number that both a and b divide. For example: $\operatorname{lcm}(6, 15) = 30$.

Problem D. A Dance with DS

Time limit 1500 ms
Mem limit 262144 kB
OS Windows

Alice and Bob are playing a game. Alice gives Bob an integer k. Then Bob picks another non-negative integer n and performs some moves (possibly zero) on n. In one move, he does the following:

- If k divides n (i.e. $n \bmod k = 0$), then he divides n by k. Formally, he does $n := \frac{n}{k}$.
- Otherwise, he subtracts 1 from n. Formally, he does $n := n - 1$.

If $n = 0$, he stops performing any further move. Bob wants to play for as long as possible, but he does not want to start with a number too big. Specifically, he doesn't want n to be bigger than a limit r. Given the limit r, what is the maximum number of moves that can be performed if he chooses n such that, $0 \le n \le r$?

Input

Input starts with an integer T ($1 \le T \le 10^5$), denoting the number of test cases.

Each of the next T lines contain two integers r ($0 \le r \le 10^{18}$) and k ($2 \le k \le 10^{18}$).

Output

For each test case, you have to print the maximum number of moves that can be performed among all n, such that $0 \le n \le r$.

Examples

Input	Output
2 5 2 4 3	4 3

Note

In the first sample, we have $r = 5$. So n cannot be bigger than 5.

- If $n = 0$, number of moves $= 0$.
- If $n = 1$, number of moves $= 1$ ($1 \to 0$).
- If $n = 2$, number of moves $= 2$ ($2 \to 1 \to 0$).
- If $n = 3$, number of moves $= 3$ ($3 \to 2 \to 1 \to 0$).
- If $n = 4$, number of moves $= 3$ ($4 \to 2 \to 1 \to 0$).
- If $n = 5$, number of moves $= 4$ ($5 \to 4 \to 2 \to 1 \to 0$).

If we choose $n = 5$, we get the maximum number of moves $= 4$.

Problem E. Winter Gifts

Time limit	1000 ms
Mem limit	262144 kB
OS	Windows

At the beginning of the winter season in 2022, Ayon and Lagno received two colorful scarves as gifts from their mother. The scarves each have their own color patterns which can be described as a string of letters where each letter identifies as a unique color. Lagno was worried that her scarf would not match her brother's, but fortunately, she found a way to modify the colors on her own scarf.

Consider the color pattern of a scarf as a string consisting of lowercase English letters only. The pattern for Lagno's scarf is S and Ayon's scarf is T. Both strings are of equal length n. For a given integer k, you can perform the following operation as many times as you want (maybe zero).

1. Choose two indices i and j such that $1 \leq i, j \leq n$ and $|i - j| = k$.
2. Change $S[i]$ to $S[j]$.

Here, $|i - j|$ is the *absolute difference* between i and j.

Can Lagno change her string S to make it equal to her brother's string T?

Input

Input starts with an integer T ($1 \leq T \leq 10^6$), denoting the number of test cases.

Each case contains three lines. The first line contains two integers n ($2 \leq n \leq 10^6$) and k ($0 < k < n$). The following two lines contain Lagno's string S and Ayon's string T. It is guaranteed that both strings have a length of n. The strings consist of lowercase English letters.

The sum of n over all test cases does not exceed 10^6.

Output

For each test case, print a single line containing "Yes" (without quotes) if string S can be made equal to T after zero or more operations, and "No" (without quotes) otherwise.

Examples

Input	Output
2 5 2 ebadc abcbc 12 1 aaabccabbdbb abaaaabdbbbc	Yes No

Note

In the first sample, you can perform the following operations:

1. $i = 1$ and $j = 3$ to make $S =$ 'abadc'.
2. $i = 3$ and $j = 5$ to make $S =$ 'abcdc'.
3. $i = 4$ and $j = 2$ to make $S =$ 'abcbc'.

Problem F. First Year, Second Year

Time limit	1000 ms
Mem limit	262144 kB
OS	Windows

Battle of Brains (BoB) is the yearly programming contest among students of the University of Dhaka. Only first and second year students of DU (any department or institute) can take part in the contest. The sole purpose of this contest is to spark an interest in the realm of problem solving in the mind of the freshers and sophomores.

In CSEDU lab, there are two computers on each table. Two contestants may communicate if they are classmates and sit at the same table. But communication is not allowed between contestants since this is an individual contest. So the organizers planned not to seat two classmates at the same table. After all the contestants arrive, the volunteers are going to seat two contestants from two different years at each table. An interesting fact is that every year the number of first-year contestants is larger than the number of second-year contestants. So after a while, some contestants are still standing there and all are first-year students.

The volunteers count the total number of contestants and the number of first-year students who are still standing there.

Now you will be given the total number of contestants and the number of first-year contestants who are still standing there. Your task is to find the number of contestants from first-year and the number of contestants from second-year.

It is guaranteed that the given data is valid according to the aforementioned scenario.

Input

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

The only line of each test case contains two integers a and b $(1 \leq b < a < 10^9)$ describing the total number of contestants and the number of first-year students who are still standing there.

Output

For each test case print a single line containing two integers separated by a space describing the number of first-year contestants and the number of second-year contestants.

Examples

Input	Output
2 12 2 15 3	7 5 9 6

Note

In the first sample, there are 7 first year students and 5 second year students. One can see that this is correct because the total number of students is 12 and there will be exactly 2 first year students still standing because 5 of them will be paired with 5 second year students.

Problem G. Rasta Thamaye Dilo

Time limit 1500 ms
Mem limit 262144 kB
OS Windows

Jubayer is a nice and talented kid. He is a two times ICPC World Finalist. He used to utilize his time solving problems after problems. But after all those years of hard work and achievements, he decides to enjoy a tiny break from his daily routine and roam to the villages of his very own B.Baria district. So, he went to B.Baria. But he soon became very fed up as he saw he cannot go to every village in his district as there are not enough roads that can help him reach every village. Also, a weird pattern was caught to the attention of this great programmer. In his district, there are $n - 1$ villages. The villages are named with natural numbers (don't ask me why, I am just a simple problem setter). For some weird reason, there is no village numbered as 1. Villages are named from 2 to n (for example: if $n = 5$ then name of the villages are 2, 3, 4 and 5). And there is a road connecting the village i and the village j **if and only if either i is divisible by j, or j is divisible by i**.

Now, Jubayer wishes to create some new roads to extend the reachability: he wants to connect every village and roam everywhere without any hassle. But he is failing to determine the minimum number of roads that he needs to construct in order to fulfill this purpose. Now it is your turn to help this world famous programmer.

Input

First line will be number of test cases, T $(1 \leq T \leq 10^5)$.

Next T lines will have a number n $(2 \leq n \leq 10^7)$ denoting the name of the last village in the district.

Output

For each test case, print the minimum number of roads needed to connect all the villages in a single line.

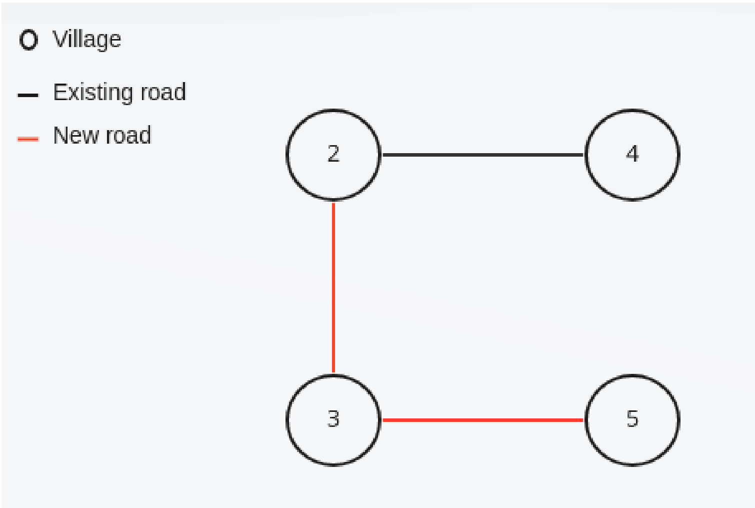
Examples

Input	Output
2 2 5	0 2

Note

Use faster input/output.

Explanation of the second test case: Here $n = 5$, so the villages are named $2, 3, 4, 5$. There's already an edge between 2 and 4 because 2 divides 4.



You have to construct at least 2 roads to make every village reachable to every other (via some roads).

Problem H. Lucky Seats

Time limit	1000 ms
Mem limit	262144 kB
OS	Windows

The FIFA World Cup 2022 will take place in Qatar. The organizers plan to provide a prize to random attendees at the opening game. So they chose two integers a and b. The integer a represents bitwise $\mathbf{OR}$ of a set of seat numbers and b represents bitwise $\mathbf{XOR}$ of the same set of seat numbers. The attendees assigned to those seats will receive the prizes. The organizers want to award prizes to as many people as possible. They therefore recruited you to identify the maximum possible number of attendees who are qualified for the award as well as their seat numbers. Each seat has a unique seat number, which is a non-negative integer.

See the notes if you forgot what bitwise $\mathbf{OR}$ and $\mathbf{XOR}$ means.

Input

Each test consists of multiple test cases. The first line contains a single integer T $(1 \le T \le 100)$ — the number of test cases.

For each testcase, there will be two numbers representing a and b $(0 \le a, b \le 10^4)$.

Output

For each testcase, if there is no solution, print **-1**.

Otherwise, print two lines.

- First line will contain the maximum possible number of attendees who will get prize.
- The next line will contain the seat numbers in increasing order separated by a single space. If there are multiple solutions for the given input, print any of them.

Examples

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

Note

Consider the first sample test. One can verify that the bitwise **OR** of the numbers $[0, 4, 5]$ is 5 , and the bitwise **XOR** of the numbers $[0, 4, 5]$ is 1. One can verify that there's no set of (distinct) numbers with a bigger size that satisfies these conditions.

Bitwise XOR: The bitwise XOR of non-negative integers A and B, denoted $A \oplus B$, is defined as follows:

- When $A \oplus B$ is written in base two, the k-th digit from the right $(k \geq 0)$ is 1 if and only if exactly one of the digits in that place of A and B is 1, and 0 otherwise. For example, we have $3 \oplus 5 = 6$ (in base two: $011 \oplus 101 = 110$).

Bitwise OR: The bitwise OR of non-negative integers A and B, denoted $A \mid B$, is defined as follows:

- When $A \mid B$ is written in base two, the k-th digit from the right $(k \geq 0)$ is 1 if and only if at least one of the digits in that place of A and B is 1, and 0 otherwise. For example, we have $3 \mid 5 = 7$ (in base two: $011 \mid 101 = 111$).

Problem I. Yet Another Mysterious Array

Time limit	1000 ms
Mem limit	262144 kB
OS	Windows

Alice and Bob went on an adventure, where they found a mysterious array A consisting of N positive integers. The i-th element of the array is called A_i.

Alice and Bob decided to play a game with this array, where each player moves in turn, and whoever can't find a valid move loses. **Alice goes first.** In simple words, Alice makes her move, then it is Bob's move, then it is Alice's move again, and so on. In every move, the player (whose move it is) does the following things sequentially:

1. Select any **prime** number p such that there exists an index i such that p divides A_i. *If no such prime number is found, the player immediately loses.*
2. For each element A_j from the array, if p divides A_j, replace A_j by $\frac{A_j}{p}$, else A_j is kept unchanged. (Note that this part should be done for every element of the array, not just one).
3. The updated array is passed to the next player.

Given the array, can you tell who will win the game if both players play optimally?

Input

The first line contains a positive integer representing the number of test cases.

For each test case, the first line contains a positive integer N and the following line contains N space-separated positive integers representing the array A.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N \leq 10^5$
- $1 \leq A_i \leq 10^5$
- **Sum over N in all test cases will not exceed** 10^5.

Output

For each test case, print "Alice" or "Bob" in a single line without the quotes referring to who will win the game.

Examples

Input	Output
2 4 2 3 2 1 5 727 86197 46727 43969 34157	Bob Alice

Note

In the first sample test case:

- Alice selects 2, the array becomes $[1, 3, 1, 1]$.
- Bob selects 3, the array becomes $[1, 1, 1, 1]$.
- Alice has no move, and Bob wins.

Problem J. Even Out

Time limit 1000 ms
Mem limit 262144 kB
OS Windows

You are given an array A of n integers, where the i-th element is A_i. You can choose exactly one i such that $1 \le i \le n$ and change A_i to $-A_i$ (in other words, you change the sign of A_i). You want to make the sum of the array even. In how many ways can you do this?

Input

The first line contains an integer $T (1 \le T \le 100)$, the number of test cases.

For each test case, there are two lines. On the first line you are given n $(1 \le n \le 100)$, the number of integers in the array. The next line contains n space-separated integers $A_1, A_2, ..., A_n$ $(-100 \le A_i \le 100)$ denoting the array A.

Output

For each test case, print one integer in seperate lines denoting the number of ways you can make the sum of the array even.

Examples

Input	Output
2 2 0 0 6 2 3 5 -2 -3 0	2 0

Note

In the first sample, there are two 0s. You can change the sign of either of them (it remains 0). So the sum is always 0, which is even. So there are 2 ways to make the sum even.

In the second sample, you can verify that no matter which element you change the sign of, the sum is not even. So there are 0 ways.

Problem K. Carrot Cakes

Time limit 1000 ms
Mem limit 262144 kB

In some game by Playrix it takes t minutes for an oven to bake k carrot cakes, all cakes are ready at the same moment t minutes after they started baking. Arkady needs at least n cakes to complete a task, but he currently don't have any. However, he has infinitely many ingredients and one oven. Moreover, Arkady can build one more similar oven to make the process faster, it would take d minutes to build the oven. While the new oven is being built, only old one can bake cakes, after the new oven is built, both ovens bake simultaneously. Arkady can't build more than one oven.

Determine if it is reasonable to build the second oven, i.e. will it decrease the minimum time needed to get n cakes or not. If the time needed with the second oven is the same as with one oven, then it is unreasonable.

Input

The only line contains four integers n, t, k, d ($1 \le n, t, k, d \le 1\ 000$) — the number of cakes needed, the time needed for one oven to bake k cakes, the number of cakes baked at the same time, the time needed to build the second oven.

Output

If it is reasonable to build the second oven, print "`YES`". Otherwise print "`NO`".

Examples

Input	Output
`8 6 4 5`	`YES`

Input	Output
`8 6 4 6`	`NO`

Input	Output
10 3 11 4	NO

Input	Output
4 2 1 4	YES

Note

In the first example it is possible to get 8 cakes in 12 minutes using one oven. The second oven can be built in 5 minutes, so after 6 minutes the first oven bakes 4 cakes, the second oven bakes 4 more ovens after 11 minutes. Thus, it is reasonable to build the second oven.

In the second example it doesn't matter whether we build the second oven or not, thus it takes 12 minutes to bake 8 cakes in both cases. Thus, it is unreasonable to build the second oven.

In the third example the first oven bakes 11 cakes in 3 minutes, that is more than needed 10. It is unreasonable to build the second oven, because its building takes more time that baking the needed number of cakes using the only oven.