Problem A. Guess The Tree

Time limit 1000 ms
Mem limit 524288 kB

Description

There is a full binary tree with n levels(so it has exactly $2^n - 1$ nodes). Each node has an integer ID from 1 to $2^n - 1$, and the $2^n - 1$ IDs form an arrangement from 1 to $2^n - 1$, but you don't know.

You need to find the IDs of the $2^n - 1$ nodes using at most 4800 queries.

Input

The first line contains one integer $n (1 \le n \le 10)$, the levels of the full binary tree.

To ask a query, you need to pick two nodes with IDs $u, v (1 \le u, v \le 2^n - 1)$, and print the line of the following form:

> "? u v"

After that, you will receive:

> "t"

The lowest common ancestor's ID of u and v.

You can ask at most 4800 queries.

If you have found the structure of the tree, print a line of the following form:

"! f_1 f_2 f_3 f_4 ... f_{2^n-1}"

f_i means the i-th node's father's ID. If it has no father, then $f_i = -1$.

After printing a query or the answer for a test case, do not forget to output the end of line and flush the output. Otherwise, you will get the verdict 'Idleness Limit Exceeded'. To do this, use:

`fflush(stdout)` or `cout.flush()` in C++;

`System.out.flush()` in Java;

`stdout.flush()` in Python.

The interactor in this task is not adaptive.

Sample 1

Input	Output
2	? 1 2
3	? 2 3
3	? 1 3
3	! 3 3 -1

Hint

In this case, the tree's root is 3, it's two sons are 1 and 2.

For any query "? a b",if $a \neq b$, the jury will return answer 3.

So we found the tree's root is 3 .

Problem B. Turn Off The Lights

Time limit 2000 ms
Mem limit 524288 kB

Description

Kitty has n^2 lights, which form an $n \times n$ matrix.

One day, Kitty found that some of these lights were on, and some were off. Kitty wants to turn them all off.

To achieve her goal, Kitty can perform three types of operations:

- (1) Choose a row, reverse the state of this row. It means if a light of this row is on, after this operation, it is now off. If a light of this row is off, after this operation, it is now on.
- (2) Choose a column, reverse the state of this column. It means if a light of this column is on, after this operation, it is now off. If a light of this column is off, after this operation, it is now on.
- (3) Choose exactly one light, reverse the state of this light. **This operation can only be performed not more than k times.**

For the current state, help Kitty achieve her goal within $3n$ operations.

Input

The first line contains two integers $n(1 \leq n \leq 1000), k(0 \leq k < n)$, indicating as described above.

Then n lines follow, each line has exactly n numbers, 0 represents that the light is turned off at this time, while 1 represents the opposite.

The y-th number of the $(x+1)$-th line in input means the light at coordinate (x, y).

Output

If Kitty can not achieve her goal,print -1 in a single line.

Otherwise, print $M(0 \leq M \leq 3n)$ in the first line, indicating the number of operations she needs to perform.

The next M lines, each line contains 2 integers x, y, separated by white space.

If $1 \leq x \leq n, 1 \leq y \leq n$, it means Kitty will reverse the light at coordinate (x, y).

If $x = 0, 1 \leq y \leq n$, it means Kitty will reverse all lights at coordinates $(z, y) 1 \leq z \leq n$.

If $1 \leq x \leq n, y = 0$, it means Kitty will reverse all lights at coordinates $(x, z) 1 \leq z \leq n$.

If there are multiple answers, print any of them.

Sample 1

Input	Output
2 0 0 1 1 0	2 0 2 2 0

Sample 2

Input	Output
3 1 1 0 0 0 1 0 0 0 1	-1

Problem C. Fix the Tree

Time limit 5000 ms
Mem limit 1048576 kB

Description

You are given a tree consisting of n vertices. Each vertex i of this tree has a value w_i assigned to it.

Now the vertex u will be broken. Once it's broken, vertex u and all edges with one end at vertex u will be removed from the tree.

To make the tree connected, you can do the following operation any number of times(possibly zero) in any order:

- First choose two vertices u and v from the tree;
- Then pay $w_u + w_v$ coins, and add an edge between vertices u and v;
- At last, replace $w_u + 1$ with w_u, replace $w_v + 1$ with w_v.

Your task is to calculate the minimum number of coins to be paid.

But you don't know which vertex u is, so you need to find the answer for each $1 \le u \le n$. Please answer all the queries independently.

Input

The first line contains a single integer $n(2 \le n \le 10^6)$ — the number of vertices in this tree.

The next line contains n numbers, the i -th number is $w_i(1 \le w_i \le n)$.

The next $n-1$ lines contain a description of the tree's edges. The i -th of these lines contains two integers u_i and $v_i(1 \le u_i, v_i \le n)$ — the numbers of vertices connected by the i -th edge.

It is guaranteed that the given edges form a tree.

Output

Print n integers, the i -th integer is the answer when $u = i$.

Sample 1

Input	Output
6 1 1 1 1 1 1 1 2 1 3 1 4 2 5 2 6	4 4 0 0 0 0

Sample 2

Input	Output
4 1 2 3 4 1 2 1 3 1 4	12 0 0 0

Sample 3

Input	Output
7 1 2 3 4 5 6 7 1 2 1 3 2 4 2 5 3 6 3 7	5 12 16 0 0 0 0

Hint

给定一个有 n 个点组成的树，每个点有一个权值 w_i。
点 u 和相邻的边被删除。
你可以进行以下操作任意次数（可以为 0）,让树重新成为连通图：

1. 选择两个点 u、v；
2. 花费 $w_u + w_v$ 的代价连接一条边 (u, v)；
3. $w_u \leftarrow w_u + 1, w_v \leftarrow w_v + 1$。

对于每个 u 计算最小代价。

Problem D. Make Them Straight

Time limit 2000 ms
Mem limit 524288 kB

Description

There is a sequence a of non-negative integers of length n, the i-th element of it is $a_i (1 \leq i \leq n)$.

A sequence is defined as 'good' if and only if there exists a non negative integer $k (0 \leq k \leq 10^6)$ that satisfies the following condition:

$a_i = a_1 + (i-1)k$ for all $1 \leq i \leq n$.

To make this sequence 'good', for each $i (1 \leq i \leq n)$, you can do nothing, or pay b_i coin to replace a_i with any non-negative integer.

The question is, what is the minimum cost to make this sequence 'good'.

Input

The first line contains an integer $n (1 \leq n \leq 2 \times 10^5)$, described in the statement.

The second line contains n integers $a_1, \ldots, a_n (0 \leq a_i \leq 10^6)$.

The third line contains n integers $b_1, \ldots, b_n (0 \leq b_i \leq 10^6)$.

Output

One integer, the answer.

Sample 1

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

Sample 2

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

Problem E. Dumb Robot

Time limit 3000 ms
Mem limit 524288 kB

Description

You have a dumb robot, and you are going to let it play games with n robots.

There is a matrix A with three rows and three columns in the game. We call the number of row i and column j of this matrix $A_{i,j}$. The game goes like this:

Two players each choose an integer from $[1,3]$ at the same time. We call the number your robot chooses i, and the number the other robot chooses j.
The score is $A_{i,j}$.
In game i, your robot will play this game with the i -th robot. The i -th robot has a probability of choosing 1 as $p_{i,1}$, a probability of choosing 2 as $p_{i,2}$, and a probability of choosing 3 as $p_{i,3}$.

Your goal is to make the expected value of the score not negative in each game. But your robot is very dumb, so it will choose 1 with probability q_1, 2 with probability q_2, and 3 with probability q_3, and you don't know the value of q_1, q_2, q_3.

We all know that $q_1 + q_2 + q_3 = 1$. If q_1, q_2, q_3 are chosen uniformly at random from a set of all possible cases, please calculate the probability that you will reach your goal.

Input

The first line contains one integer $n (1 \leq n \leq 10^4)$.

Each of the next 3 lines contains 3 integers, the j -th integer in the i -th of these lines is $A_{i,j} (-20 \leq A_{i,j} \leq 20)$.

Each of the next n lines contains 3 real numbers, the j -th number in the i -th of these lines is $p_{i,j}$. It is guaranteed that $p_{i,1} + p_{i,2} + p_{i,3} = 1$ and $0 \leq p_{i,j}$.

Output

Output the answer to the problem. It is guaranteed that the answer will never be 0.

Your answer is considered correct if its absolute or relative error does not exceed 10^{-2}. Formally, let your answer be a, and the jury's answer be b. Your answer is accepted if and only if $\frac{|a-b|}{max(1,|b|)} \le 10^{-2}$.

Sample 1

Input	Output
1 1 1 1 -1 2 1 0 -3 2 0.1 0.6 0.3	0.748252

Sample 2

Input	Output
8 1 3 -2 0 0 2 -2 2 1 0.1 0.3 0.6 0 0 1 0.5 0.2 0.3 0 0 1 1 0 0 0 0 1 0.33 0.33 0.34 0.16 0.16 0.68	0.111111

Hint

In example 1, for example, $(q_1 = 1, q_2 = 0, q_3 = 0)$ is ok. In this case, Your robot will always choose 1, so no matter what number will robot 1 choose, the score will always be 1, which is enough to reach your goal.

Problem F. XOR Game

Time limit 1000 ms
Mem limit 524288 kB

Background

statement updated:

z is the number of numbers whose values are 0.

Description

Alice and Bob are playing a game against each other.

In front of them are a multiset $\{a_i\}$ of non-negative integers and a single integer x. Each number in a is 0 or $2^i (0 \le i < k)$ before the game.

This game will be a turn-based game, and Alice will go first. In one person's turn, he or she will choose an integer from a. Let this number be p. Then this person will choose whether or not to make $x \leftarrow x \oplus p$, then remove p from a. Here, operation $\oplus$ means bitwise xor.

Alice wants to make x as big as possible, and Bob wants to make x as small as possible.

You are a bystander who wants to know the final value of x. However, the size of a is a huge number. Formally, there are b_i numbers whose values are 2^i in a for all $0 \le i < k$, and z numbers whose values are 0. But you still want to challenge this impossible problem.

If Alice and Bob are smart enough, please output the final value of x.

Input

The first line contains two integers $k, z (1 \le k \le 10^5, 0 \le z \le 10^9)$.

The next line contains k integers, the i -th integer is $b_{i-1} (0 \le b_{i-1} \le 10^9)$.

Output

Output the answer in binary format. Note that you should output exactly k digit from high to low even though this number has leading 0s.

Sample 1

Input	Output
1 0 3	1

Sample 2

Input	Output
2 0 2 1	11

Sample 3

Input	Output
2 0 2 2	00

Problem G. The Last Cumulonimbus Cloud

Time limit 1000 ms
Mem limit 524288 kB

描述

每年四月，这座城市总是被积雨云笼罩。

这座城市由n栋建筑和m条双向街道连接。为了方便人们的出行，任意两座建筑物都可以通过街道直接或间接到达彼此。同时，没有街道连接同一座建筑物，并且每对建筑物之间最多有一条街道相连。

这座城市的生活节奏非常缓慢，因为城市布局并不是很庞大。

具体来说，如果我们将这座城市视为一个无向图G，则保证对于图中的任何非空子图，都至少有一座建筑物在其中，该建筑物与子图内最多连接10条街道。

雨势不停，积雨云的数量不断增加。一开始，在第i座建筑物上方有a_i朵积雨云，但在接下来的q天里，每天将发生以下两种事件之一：

- 在第x座建筑物上方增加v朵积雨云。
- 2 x你需要报告与第x座建筑物直接相连的所有建筑物上总共有多少朵积雨云。

输入

第一行包含三个整数$n, m, q (1 \le n \le 3 \times 10^5, 1 \le m \le 3 \times 10^6, 1 \le q \le 2 \times 10^6)$。

接下来的每一行包含两个整数$x, y (1 \le x, y \le n, x \neq y)$，表示连接第$x$座和第$y$座建筑物的街道。

接下来的每一行包含一个整数$a_i (0 \le a_i \le 100)$。

接下来的每一行包含两个或三个整数，如果第一个整数是1，表示第一种事件，接下来两个整数表示$x, v (0 \le v \le 100)$。如果第一个整数是2，表示第二种事件，接下来一个整数表示x。

输出

多行，每行代表一个第二种事件的查询结果。

示例 1

Input	Output
4 6 10 2 4 2 3 4 3 3 1 4 1 2 1 0 7 1 6 2 4 2 2 1 3 3 2 1 1 1 9 2 4 2 2 1 3 6 2 4 2 2	8 7 17 20 19 26 25

Problem H. Holes and Balls

Time limit 3000 ms
Mem limit 524288 kB

Description

You are given n balls, the i -th ball's value is p_i. It's guaranteed that $p_1, p_2, \ldots, p_n$ is a permutation of $1, 2, 3 \ldots, n$.

There is also a rooted tree of n vertices, each of the vertices is a hole, and each hole can only hold one ball.

The tree's root is the first vertex.

Now you need to fill the holes with the balls.

You need to throw each ball in order of 1 to n in the following steps:

1. Throw the ball into vertex 1.
2. Let the vertex where the ball is be p.
3. If the p -th vertex has already been filled with other balls, you need to choose a vertex x and throw the ball into the x -th vertex, then return to step 2. You need to guarantee that the x -th vertex is the p -th vertex's son and at least one vertex in the subtree of the x -th vertex is not filled.
4. Otherwise, the ball will fill the p -th vertex.

After throwing all the balls, let a_i express the value of the ball in the i -th vertex.

You need to find the minimum lexicographical order of $\{a_n\}$.

We define dep_i as the number of vertices on the path from the i -th vertex to the tree's root(the first vertex).

Specially, for any two vertices $x < y$, it's guaranteed that $dep_x \le dep_y$.

Input

The first line contains a single integer $n (1 \le n \le 5 \times 10^5)$ - the number of vertices in this tree.

The next line contains n numbers, the i -th number is $p_i(1 \le p_i \le n)$. It's guaranteed that $p_1, p_2, \dots, p_n$ is a permutation of $1, 2, 3 \dots, n$.

The next $n-1$ lines contain a description of the tree's edges. The i -th of these lines contains two integers u_i and $v_i(1 \le u_i, v_i \le n)$ - the numbers of vertices connected by the i -th edge.

It is guaranteed that the given edges form a tree.

And for any vertices $x < y$, it's guaranteed that $dep_x \le dep_y$.

Output

Output n integers, the minimum lexicographical order of $\{a_n\}$.

Sample 1

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

Sample 2

Input	Output
9 9 2 6 3 5 7 1 4 8 1 2 1 3 2 4 2 5 3 6 3 7 4 8 4 9	9 2 1 3 6 4 8 5 7

Problem I. Smart Quality Inspector

Time limit 500 ms
Mem limit 524288 kB

Description

Ella has a factory. One day, her factory is facing a product quality inspection.

Her factory has N production lines. Among the N production lines, $N-K$ of them are qualified, and the other K lines are unqualified. The fine of the i -th$(1 \le i \le K)$ unqualified line is i Yuan.

There are M quality inspectors here. For the j -th$(1 \le j \le M)$ quality inspector, he will inspect from the l_i -th line to the r_i -th line and find the unqualified production line with the largest fine among them, then impose this fine on Ella.

Ella does not want to receive so many fines, so she decides to renumber the N production lines to receive the least amount of fines. Please help her.

In simple terms:

You have a sequence A of length N, $A = [1, 2, 3, ..., K, 0, 0, 0, ..., 0]$. Here N, K are given.

There are M pairs of integers, each pair consists of two numbers l_i, r_i.

You need to rearrange sequence A to minimize the following:

$$\sum_{i=1}^{M} \max_{j=l_i}^{r_i}(A_j)$$

Input

The first line contains three integers $N, K, M (1 \le K \le N \le 20, 1 \le M \le 10^5)$, described in the statement.

Then M lines, the i -th line of them contains two integers $l_i, r_i (1 \le l_i \le r_i \le N)$.

Output

An integer indicates the answer.

Sample 1

Input	Output
4 4 3 1 2 3 4 1 4	10

Problem J. Triangle

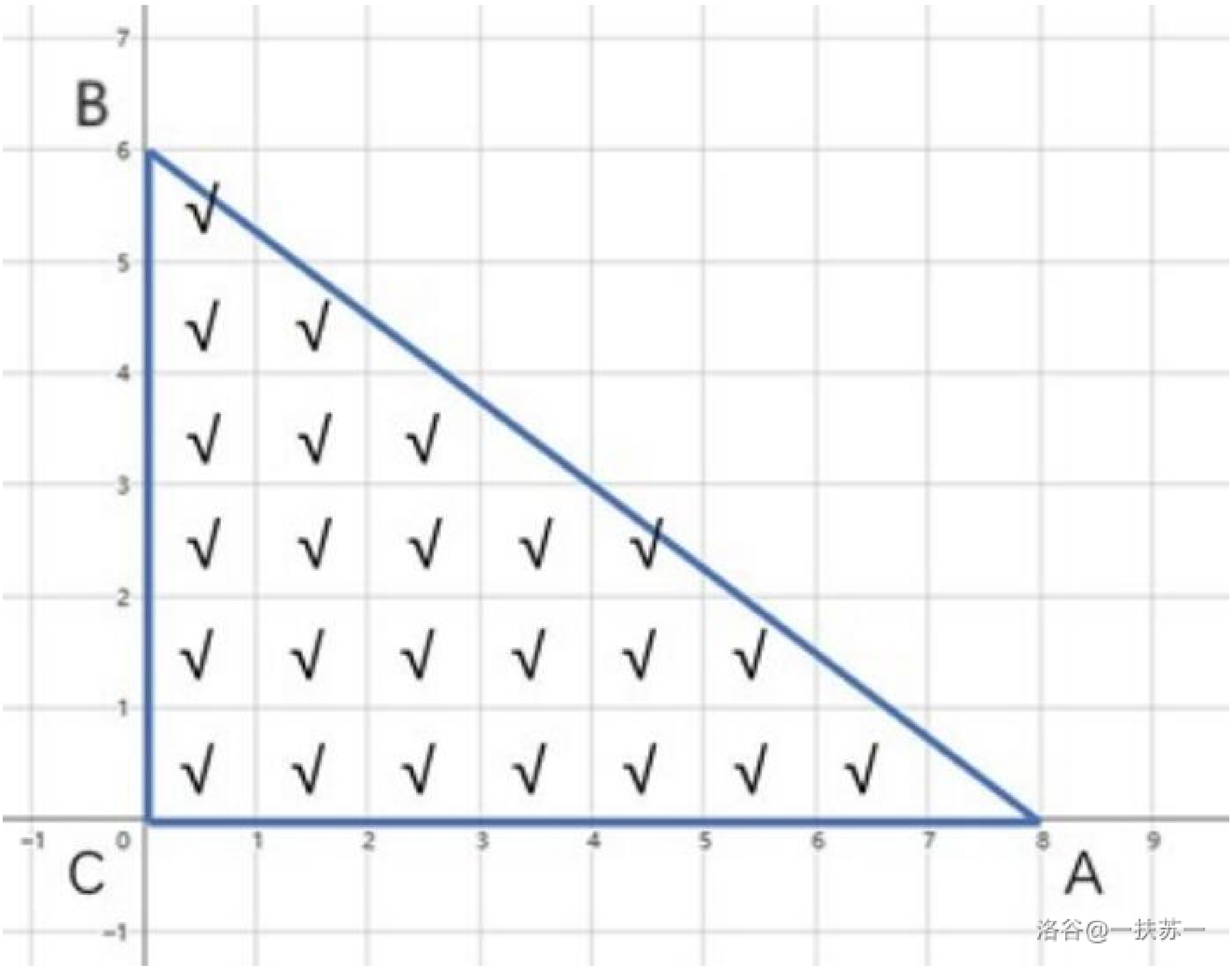
Time limit 1000 ms
Mem limit 524288 kB

Description

There are three points $A(a, 0)$, $B(0, b)$, $C(0, 0)$ in the plane rectangular coordinate system. Define the size of triangle ABC as the number of squares that belong to it.

For integers x, y, one square is defined by four points (x, y), $(x + 1, y)$, $(x, y + 1)$, $(x + 1, y + 1)$. We consider a square to belong to a triangle if half or more of it is inside the triangle.

For example, when $a = 8, b = 6$, the size of triangle ABC will be 24. The image below shows which squares are counted in the triangle.



Input

The first and only line contains two integers a $(1 \le a \le 10^6)$ and b $(1 \le b \le 10^6)$.

Output

Print an integer: the size of triangle ABC.

Sample 1

Input	Output
6 8	24

Sample 2

Input	Output
5 5	15

Sample 3

Input	Output
1 999999	500000

Problem K. Yet Another Maximum Matching Counting Problem

Time limit 3000 ms
Mem limit 524288 kB

Description

There is a two-dimensional plane.

You have a set of points $\{(x_i, y_i)\}$ that satisfies $1 \le x_i \le n, 1 \le y_i \le m$ (Both x_i and y_i are integers), and there are no two points with the same coordinates.

If two points have the same horizontal or vertical coordinates, we will connect an edge between these two points. This forms a graph.

You need to find the sum of the maximum number of matches in the graphs formed by all possible $2^{nm} - 1$ non empty sets, and output the result modulo 998244353.

Here, the maximum number of matches in a graph is defined as: selecting the most edges so that there are no common vertices between any two edges.

Input

There are multiple testcases in this problem.

The first line contains an integer $T (1 \le T \le 100)$, which represents the number of testcases.

Each of the testcases contains two integers $n, m (1 \le n, m \le 500)$.

Output

For each of the testcases, print an integer representing the result modulo 998244353.

Sample 1

Input	Output
10 1 1 1 2 2 2 4 4 3 3 5 5 1 8 20 20 100 100 500 500	0 1 10 241456 964 200419152 448 985051144 370696900 357517517

Problem L. Rubbish Sorting

Time limit 2000 ms

Mem limit 1048576 kB

Description

Bob has many pieces of rubbish. One day, he wants to sort them.

For every piece of rubbish, its type is expressed as a positive integer.

He has q operations. For each operation, it is one of the following two operations.

- `1 s x` He tells you that the piece of rubbish named s has a type of x.
- `2 s` He wants to ask you the type of rubbish s.

But his memories are not always accurate.

For each operation 2, s may not have appeared in the previous operation 1s.

We define the similarity of two strings s_1 and s_2 as $\sum_{i=1}^{\min\{|s_1|,|s_2|\}}[s_{1,i} = s_{2,i}]$.

Here all the strings' indexes start at 1.

For a string s, its type is the type of string which has the maximum similarity to s among all the strings that have appeared in the previous operations 1s. Note that if there are multiple strings that all have the maximum similarity to s, the type of s is the minimum type of these strings' type.

Now, he wants you to solve this problem.

Input

The first line contains an integer $q(1 \le q \le 3 \times 10^5)$, which is the number of operations.

Next q lines contain operations, one per line. They correspond to the description given in the statement.

It is guaranteed that for every operation 2 there is at least one operation 1 before it.

But some pieces of rubbish will have more than one type, you can consider it as the minimum type you have read.

The rubbish's names only consist of lowercase Latin letters.

$1 \le |s| \le 5, 1 \le x \le 10^9$

Output

For every operation 2, you should print an integer in a single line that is the rubbish s's type.

Sample 1

Input	Output
4 1 aaa 1 2 aa 1 ab 2 2 bb	1 2

Sample 2

Input	Output

Problem M. Chained Lights

Time limit 1000 ms
Mem limit 1048576 kB

Description

You have n lights in a row. Initially, they are all off.

You are going to press these n lights one by one. When you press light i, light i will switch its state, which means it will turn on if it's off and turn off if it's on, and then for every j satisfied $i|j, i < j \leq n$, press light j once.

For example, if $n = 4$, when you press light 1, light 1 will turn on, and then you will press light 2, 3, 4. Since you pressed light 2, light 2 will turn on and you will press light 4, which will cause light 4 to turn on. After all the operations, lights 1, 2, 3 will be turned on and light 4 is still off.

You will press these n lights and do the operations as mentioned above one by one. After all the operations, you want to know whether light k is on or off.

You can also use the following code to understand the meaning of the problem:

```
void press(int x)
{
    light[x]^=1;
    for (int y=x+x;y<=n;y+=x) press(y);
}
for (int i=1;i<=n;i++) press(i);
```

Input

There are multiple testcases.

The first line contains an integer $T(1 \leq T \leq 10^5)$, which represents the number of testcases.

Each of the testcases contains two integers $n, k(1 \leq k \leq n \leq 10^6)$ in a single line.

Output

For each testcase, if light k is turned on in the end, output YES, otherwise output NO.

Sample 1

Input	Output
2 1 1 3 2	YES NO