Problem A. Play with Koko

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Orange`

Koko loves cats and has a lot of them. Recently, He found it difficult to name all of them.

He found a way to solve this by giving each cat an **ID** ($2 \leq$ ID), Which is the product of its parent's IDs.

Now, Koko wants to play with you, he will give you two cats' IDs, and you have to **determine** if they have some common ancestor.

Input

The input begins with a positive integer ($1 \leq t \leq 100$), the number of test cases.

After that, follows t lines, each with two integers a_i , b_i identifying two cats' IDs. ($2 \leq a_i, b_i \leq 1000$)

Output

For each test case, print "YES"if the cats a_i and b_i share a common ancestor and "NO"otherwise.

Example

standard input	standard output
2 2 4 3 5	YES NO

Problem B. Coprime Trees

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Red`

Given n, l, and r, count the number of trees rooted at 1 which satisfy the following rules:

1. The number of nodes doesn't exceed n.
2. The values of nodes are between l and r.
3. The greatest common divisor between any pair of nodes that lie on the same path from the root is equal to 1.
4. Direct children of the same parent should have distinct values.

Two trees are considered distinct if they are not isomorphic, or if they are isomorphic but at least one of the nodes values is different.

Print the number of such trees modulo $10^9 + 7$.

Input

The first line of the input contains a single integer t ($1 \le t \le 50$) — the number of test cases. Then t test cases follow.

Each test case contains three integers n, l, and r ($1 \le n \le 55, 1 \le l, r \le 10^{18}, r - l + 1 \le 30$) — the maximum number of nodes, and the values limits.

The sum of n over all test cases does not exceed 55.

Output

For each test case, print a single integer — the number of distinct trees which satisfy the rules modulo $10^9 + 7$.

Example

standard input	standard output
2 1 1 10 2 3 3	10 1

Problem C. Yet Another Game

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Dark Blue`

Alice and **Bob** are playing a game that progresses turn by turn and as usual, Alice moves first.

There are n piles, The i_{th} pile contains a_i stones. On each turn, a player can choose a non-empty pile and remove from it any number of stones from L stones up to R stones (The pile's size should become non-negative). The player who can't make a move losses.

You are asked to find out who will win the game if both players play optimally, and find the maximum possible number of stones Alice can remove in the first turn such that the result will be the same.

Input

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

The first line of each test case will contain three integers n, L, and R ($1 \leq n \leq 10^5$), ($1 \leq L \leq R \leq 10^5$), ($L \leq a_n$).

The second line will contain n integers $a_1 \leq a_2 \leq \ldots \leq a_n$ where a_i denotes the number of stones in the i_{th} pile.

It's guaranteed that the sum of a_n overall test cases won't exceed 10^6 and the same for n.

Output

For each test case, print in a single line the winner's name and the maximum possible number of stones Alice can remove in the first turn such that the result will be the same.

Example

standard input	standard output
3 1 1 5 5 1 1 4 5 2 1 3 4 5	Alice 5 Bob 4 Alice 3

Problem D. Guess Number

Input file:	`standard input`
Output file:	`standard output`
Balloon Color:	`Yellow`

We have a hidden number N, where $1 \leq N \leq 10^9$, Your is task to find it.

You guess a number X and we respond with three types of responses, "$>$" if $N > X$, "$<$" if $N < X$, or "$=$". if the response is N equals X , which means your guess was correct and then, you shout print "! N" and terminate the program immediately.

If you make more than 30 guesses, you will receive a Wrong Answer verdict.

Interaction Protocol

After printing a number X do not forget to output end of line and flush the output. Otherwise, you will get `Idleness limit exceeded`. To do this, use:

- `fflush(stdout)` or `cout.flush()` in C++;
- `System.out.flush()` in Java;
- `flush(output)` in Pascal;
- `stdout.flush()` in Python;
- see documentation for other languages.

Example

standard input	standard output
< > > =	11 6 9 10 ! 10