1. **Climbing Down**

Two friends Walter and Jesse are currently on floors **A** and **B** respectively. They hear an announcement that prizes are being distributed on the ground floor and so decide to reach the ground floor as soon as possible.

Walter can climb down **X** floors per minute while Jesse can climb down **Y** floors per minute. Determine who will reach the ground floor first. In case both reach the ground floor together, print Both.

Input Format:
The only line of input contains four space-separated integers **A**, **B**, **X**, and **Y** - the current floor of Walter, the current floor of Jesse, speed of Walter and speed of Jesse in floors per minute respectively.

Output Format:

Print **Walter** if Walter reaches the ground floor first. **Jesse** if he reaches the ground floor first. **Both** if both reach the ground floor at the same time.

Sample I/O:

Input 1:
2 2 2 2
Output 1:
Both

Input 2:
4 2 1 5
Output 2:
Jesse

Input 3:
3 2 4 1
Output 3:
Walter

Input 4:
10 6 6 4
Output 4:
Jesse

2. **Can be a factor?**

Madmax has two integers A and B (A ≤ B).

He can choose any **non-negative** integer X and add it to both A and B. Find whether it is possible to make **A a divisor of B**.

See the Sample I/O and Explanation for more clarity.

Input Format:

- The first line of input will contain a single integer T, denoting the number of test cases.
- Each test case consists of two integers A and B.

Output Format:

For each test case, output **YES** if it is possible to make A a factor of B, **NO** otherwise.

Constraints:

- 1 ≤ T ≤ 105
- 1 ≤ A ≤ B ≤ 109

Sample I/O:

Input 1:
3
3 6
4 14
9 11
Output 1:
YES
YES
NO

Input 2:
5
7 16
21 27
16 33
10 10
130 259
Output 2:
YES
NO
YES
YES
NO

Explanation:
In Input 1,
for Testcase 1, we can choose X = 0 and can add it to both 3 and 6. Now 3 is a factor of 6.
for Testcase 2, we can choose X = 1 and can add it to both 4 and 14 making them 5 and 15. 5 is a factor 15.
for Testcase 3, there not possible value of X to add such that A becomes a factor of B.

3. Divisibility Problem

You are given two positive integers ***a*** and ***b***. In one move you can increase ***a*** by 1
(replace ***a*** with ***a+1***).
Your task is to **find the minimum number of moves you need to do in order to make *a* divisible by *b***. It is possible, that you have to make 0 moves, as ***a*** is already divisible by ***b***.

Note: Check out the Sample I/O for more clarity.

Input Format:
The only line of input contains two integers ***a*** and ***b***.

Output Format:
Print the answer: the minimum number of moves you need to do in order to make ***a*** divisible by ***b***.

Constraints:
1 ≤ a, b ≤ 1000000000

Sample I/O:

Input 1:
10 4
Output 1:
2

Input 2:
13 9
Output 2:
5

Input 3:
100 13
Output 3:
4

Input 4:
123 456
Output 4:
333

Input 5:
92 46
Output 5:
0

Explanation:
For Input 1,
Move1: You can add 1 to ***a (10)*** to make it 11.
Move2: As 11 is not divisible by ***b (4)***, we will add 1 more to ***a***, thus it becomes 12.
As 12 is divisible by ***b (4)***, we can stop at this point. We can say that minimum number of moves we have to make for this input is 2.

For Input 3,
We have to add 1 to ***a (100)*** for 4 times to get to 104, which is divisible by ***b (13)***.