K-Divisors

The *positive divisor function* is defined as a function that counts the number of positive divisors of an integer **N**, including **1** and **N**.

If we define the positive divisor function as **D(N)**, then, for example:

D(1) = 1

D(2) = 2

D(10) = 4

D(24) = 8

Calculating **D(N)** is a classical problem and there are many efficient algorithms for that. But what if you are asked to find something different? Given a range and an integer **K**, can you find out for how many **N** in the given range, **D(N)** equals **K**?

Input

In the very first line, you'll have an integer called **T**. This is the number of test cases that shall follow. Every test case contains three integers, **L**, **R**, and **K**. **L** and **R** represent the range and are inclusive.

Constraints

- $1 \leq T < 31$
- $1 \leq L \leq R < 2^{31}$
- $1 \leq K < 2^{31}$

Output

For every test case, you must print the case number, followed by the count of numbers with exactly **K** divisors in the range.

Sample Input

```
3
10 10 4
2 13 2
100 10000 100
```

Sample Output

```
Case 1: 1
Case 2: 6
```

Case 3: 0