Pure Number

Time Limit : 1 Second
Memory Limit : 512 MB

Let's start with some basic definitions. A **prime number** is a positive integer greater than 1 whose divisor are 1 and itself, e.g., 2, 3, 5, 7. Any other positive integers greater than 1 which are not prime are called **composite number**, e.g., 4, 6, 8, 9, 10.

Now, let's define a pure number. A **pure number** of degree K is a positive integer which can be formed by multiplying exactly K unique prime numbers. As you might have noticed, a pure number of degree 1 is also a prime number. For example,

- 2 is a pure number of degree 1, i.e. 2;
- 6 is a pure number of degree 2, i.e. 2 × 3;
- 35 is a pure number of degree 2, i.e. 5 × 7;
- 66 is a pure number of degree 3, i.e. 2 × 3 × 11;
- 12 is not a pure number as it cannot be formed by multiplying unique prime numbers.

By definition, 1 is not a pure number.

Given three integers L, R, and K, your task in this problem is to determine how many pure number of degree K between L and R (inclusive).

Input

Input begins with an integer: T ($1 \le T \le 100$) denoting the number of cases.

Each case contains the following input block: Each case contains three integers: L R K ($1 \le L \le R \le 10^6$; $1 \le K \le 1000$) in a line.

Output

For each case, output in a line "`Case #X: Y`" where `X` is the case number (starts from 1) and `Y` is the output for the respective case.

Examples

input	**Example #1**

`3`
`1 10 1`
`1 10 2`
`10 14 3`	

output	

```
Case #1: 4
Case #2: 2
Case #3: 0
```

explanation

Case 1: 2, 3, 5, and 7 are pure numbers of degree 1

Case 2: 6 and 10 are pure numbers of degree 2.

Case 3: There is no pure number of degree 3 between 10 and 14.

End of Problem