Problem C: Dirichlet's Theorem

Source: `dirichlet.{c,cpp,java}`

Dirichlet's theorem on arithmetic progressions states that for any two positive integers **a** and **b**, if **gcd(a,b) = 1** then the arithmetic progression **t(n) = a*n + b (n ≥ 0)** contains infinitely many prime numbers. Recall that a prime number is a positive integer ≥ 2 that has no divisors other than 1 and itself.

For example, if **a = 4** and **b = 3**, then the arithmetic progression is

3, 7, 11, 15, 19, 23, 27, 31, 35, ...,

and it can be seen that many prime numbers are contained in the first part of this list.

Given arbitrary integers **a > 0**, **b ≥ 0**, and **U ≥ L ≥ 0**, your job is to count how many values of **t(n) = a*n+b** are prime, where **L ≤ n ≤ U**.

Input

The input consists of a number of cases. The input for each case is specified by the four integers **a**, **b**, **L**, and **U** on a line. You may assume that **a*U+b ≤** 10^{12} and **U - L ≤** 10^{6}. A line containing a single 0 indicates the end of input.

Output

For each test case, print:

```
Case xxx: yyy
```

where **xxx** is the case number (starting from 1), and **yyy** is the number of **t(n), L ≤ n ≤ U**, that are prime.

Sample Input

```
4 3 0 8
1 0 2 100
2 7 0 1000
0
```

Sample Output

```
Case 1: 6
Case 2: 25
Case 3: 301
```