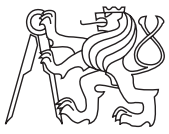

CTU Open Contest 2004

C Looooops

`c.p`, `c.c`, `c.C`

A Compiler Mystery: We are given a C-language style `for` loop of type

```
for (variable = A; variable != B; variable += C)
  statement;
```

I.e., a loop which starts by setting *variable* to value A and while *variable* is not equal to B, repeats *statement* followed by increasing the *variable* by C. We want to know how many times does the *statement* get executed for particular values of A, B and C, assuming that all arithmetics is calculated in a k-bit unsigned integer type (with values $0 \leq x < 2^k$) modulo 2^k.

Input Specification

The input consists of several instances. Each instance is described by a single line with four integers A, B, C, k separated by a single space. The integer k ($1 \leq k \leq 32$) is the number of bits of the control variable of the loop and A, B, C ($0 \leq A, B, C < 2^k$) are the parameters of the loop.

The input is finished by a line containing four zeros.

Output Specification

The output consists of several lines corresponding to the instances on the input. The i-th line contains either the number of executions of the *statement* in the i-th instance (a single integer number) or the word `FOREVER` if the loop does not terminate.

Sample Input

```
3 3 2 16
3 7 2 16
7 3 2 16
3 4 2 16
0 0 0 0
```

Output for Sample Input

```
0
2
32766
FOREVER
```