An algorithm given by Lothar Collatz produces sequences of integers, and is described as follows:

Step 1: Choose an arbitrary positive integer A as the first item in the sequence.

Step 2: If $A = 1$ then stop.

Step 3: If A is even, then replace A by $A/2$ and go to step 2.

Step 4: If A is odd, then replace A by $3 * A + 1$ and go to step 2.

It has been shown that this algorithm will always stop (in step 2) for initial values of A as large as 10^9, but some values of A encountered in the sequence may exceed the size of an integer on many computers. In this problem we want to determine the length of the sequence that includes all values produced until either the algorithm stops (in step 2), or a value larger than some specified limit would be produced (in step 4).

Input

The input for this problem consists of multiple test cases. For each case, the input contains a single line with two positive integers, the first giving the initial value of A (for step 1) and the second giving L, the limiting value for terms in the sequence. Neither of these, A or L, is larger than 2,147,483,647 (the largest value that can be stored in a 32-bit signed integer). The initial value of A is always less than L. A line that contains two negative integers follows the last case.

Output

For each input case display the case number (sequentially numbered starting with 1), a colon, the initial value for A, the limiting value L, and the number of terms computed.

Sample Input

```
3 100
34 100
75 250
27 2147483647
101 304
101 303
-1 -1
```

Sample Output

```
Case 1: A = 3, limit = 100, number of terms = 8
Case 2: A = 34, limit = 100, number of terms = 14
Case 3: A = 75, limit = 250, number of terms = 3
Case 4: A = 27, limit = 2147483647, number of terms = 112
Case 5: A = 101, limit = 304, number of terms = 26
Case 6: A = 101, limit = 303, number of terms = 1
```