Three N plus 1

Consider an algorithm that takes in a number, and processes it as follows:

If the number is odd, it multiplies it by 3, and adds 1.

If it is even, it divides it by 2.

The algorithm then repeats over this new number, until the number is equal to 1.

Thus, for example, given the number 22, the algorithm will yield the following sequence of numbers:

```
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
```

It is conjectured that this algorithm will terminate for any integral input value. Despite its simplicity, nobody knows whether this conjecture is actually true.

Now, given an input number n, the cycle length for n is the length of the sequence obtained until 1 is reached. The cycle length of 22 is therefore 16.

Your task is to determine the maximum cycle length over all numbers between some integers i and j.

Input

As input, your program will be given a pair of integers, passed in on STDIN. You should process all pairs of integers and for each pair determine the maximum cycle length over all integers between and including i and j. You can assume that no operation overflows a 32-bit integer.

For example, if your program is called tnpo.py, it will be called as

python3 tnpo.py

and then receive the following input which can be read via the following commands.

```
l=sys.stdin.readlines()
start=int(l[0])
end=int(l[1])
```

Output

Your program must print out, on an individual line, the two input numbers, as well as the maximum cycle length. Thus, for example, if your code was run with the parameters 1 100, the output should be

1 100 119

YOUR CODE SHOULD TAKE NO MORE THAN 3 SECONDS TO RUN FOR INPUT PARAMETERS 1 1000000