The 3n + 1 problem — X86177_en

Consider the following algorithm:
1. input n
2. print n
3. if $n = 1$ then STOP
4. if n is odd then $n = 3n + 1$
5. else $n = n/2$
6. GOTO 2

Given the input 22, the following sequence of numbers will be printed: 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
Given an input n, it is possible to determine the number of numbers printed before **and including** the 1 is printed. For a given n this is called the *cycle-length of n*. In the example above, the cycle length of 22 is 16.
For any two numbers i and j you are to determine the maximum cycle length over all numbers between **and including** both i and j.

Input

The input will consist of a series of pairs of integers i and j, one pair of integers per line. All integers will be less than 10,000 and greater than 0. 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.

Output

For each pair of input integers i and j you should output i, j, and the maximum cycle length for integers between and including i and j. These three numbers should be separated by at least one space with all three numbers on one line and with one line of output for each line of input. The integers i and j must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same line).

Sample input

```
1 10
100 200
201 210
900 1000
```

Sample output

```
1 10 20
100 200 125
201 210 89
900 1000 174
```

Problem information

Author : Prof. Carlos de Salles, DEINF/UFMA.
Generation : 2018-08-29 15:36:15