The 3n+1 Problem(3n+1.cpp)

https://vjudge.net/problem/UVA-100

Problems in Computer Science are often classified as belonging to a certain class of problems (e.g., NP, Unsolvable, Recursive). In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.

Consider the following algorithm:

```
1.  input $n$
2.  print $n$
3.  if $n = 1$ then STOP
4.      if $n$ is odd then $n \longleftarrow 3n + 1$
5.      else $n \longleftarrow n/2$
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

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers n such that $0 < n < 1,000,000$ (and, in fact, for many more numbers than this.)

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(3n+1.in)

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

Sample Output(3n+1.out)

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

The 3n+1 Problem
(3n+1.cpp)

[问题描述]

考虑如下的序列生成算法：从整数 n 开始，如果 n 是偶数，把它除以 2；如果 n 是奇数，把它乘 3 加 1。用新得到的值重复上述步骤，直到 n = 1 时停止。例如，n = 22 时该算法生成的序列是：

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

人们猜想（没有得到证明）对于任意整数 n，该算法总能终止于 n = 1。这个猜想对于至少 1000000 内的整数都是正确的。

对于给定的 n，该序列的元素（包括 1）个数被称为 n 的循环节长度。在上述例子中，22 的循环节长度 为 16。输入两个数 i 和 j，你的任务是计算 i 到 j（包含 i 和 j）之间的整数中，循环节长度的最大值。

[输入]

输入每行包含两个整数 i 和 j。所有整数大于 0，小于 1 000 000。

[输出]

对于每对整数 i 和 j，按原来的顺序输出 i 和 j，然后输出二者之间的整数中的最大循环节长度。这三个整数应该用单个空格隔开，且在同一行输出。对于读入的每一组数据，在输出中应位于单独的一行。

[样例输入](3n+1.in)

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

[样例输出](3n+1.out)

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