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 $n \leftarrow 3n + 1$
5. else $n \leftarrow n/2$
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

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

Sample Output

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

컴퓨터 과학의 문제들은 종종 특정 등급의 문제(예: NP, 해결 불가능, 반복적)에 속하는 것으로 분류된다. 이 문제에서 당신은 가능한 모든 입력에 대해 분류가 알려지지 않은 알고리즘의 속성을 분석하게 될 것이다.
다음 알고리즘을 고려하십시오.
1.입력 n
2. 인쇄 n
3. n = 1이면 중지
4.
←–
n이 홀수일 경우 n 3n + 1
5.
←–
다른 n/2
6. 갓도 2
입력 22를 고려할 때 다음과 같은 일련의 숫자가 인쇄된다.
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1
일체형 입력 값에 대해 위의 알고리즘이 종료될 것으로 추측된다(1을 인쇄할 때). 알고리즘의 단순성에도 불구하고 이 추측이 사실인지는 알 수 없다. 그러나 0 < 1,000, 000 (그리고 사실 이보다 더 많은 숫자에 대해서는) 모든 정수 n에 대해 검증되었다.
입력 n을 고려할 때, 인쇄 전 1을 포함한 숫자의 수를 결정할 수 있다. 주어진 n에 대해 이것을 n의 사이클 길이라고 한다. 위의 예에서 22의 사이클 길이는 16이다.
i와 j의 경우 i와 j 사이의 모든 숫자에 걸쳐 최대 사이클 길이를 결정해야 한다.

입력

입력은 한 줄에 한 쌍의 정수 i와 j의 쌍으로 구성된다. 모든 정수는 10,000보다 작고 0보다 클 것이다.
당신은 모든 정수 쌍을 처리해야 하며 각 쌍에 대해 i와 j를 포함한 모든 정수에 대한 최대 주기 길이를 결정한다.
32비트 정수를 오버플로하는 작업이 없다고 가정할 수 있다.

산출량

i와 j의 각 입력 정수 쌍에 대해 i와 j 사이의 정수에 대한 최대 사이클 길이를 출력해야 한다. 이 세 번호는 한 줄에 세 개 숫자 모두와 각 입력 라인에 대해 한 줄의 출력 라인으로 최소한 한 칸의 공간으로 구분되어야 한다. 정수 i와 j는 입력에 나타난 것과 동일한 순서로 출력에 나타나야 하며 최대 사이클 길이(동일한 라인에서) 뒤에 와야 한다.

샘플 입력
1 10
100 200
201 210
900 1000

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