

Hailstone

The following algorithm produces what is known as the hailstone sequence:

Pick some positive integer and call it n .

If n is even, divide it by two.

If n is odd, multiply it by three and add one.

Continue this process until n is equal to one.

For any given input value of n , the numbers will go up and down, but eventually—at least for all numbers that have ever been tried—comes down to end in 1. In some respects, this process is reminiscent of the formation of hailstones, which get carried upward by the winds over and over again before they finally descend to the ground.

Because of this analogy, this sequence of numbers is usually called the *Hailstone sequence*, although it goes by many other names as well. Call the length of this sequence the *Hailstone length*.

Your problem is to calculate the maximum hailstone length between two given numbers. Report only the input value that produces the maximum output.

Input

The first line of input will give the number of test cases (a positive integer, $1 \leq t \leq 100$). Each t successive lines will have two integers, $1 \leq a \leq 100,000,000$ and $1 \leq b \leq 100,000,000$, that represent the range from which to calculate the maximum hailstone length.

Output

The value of n that has the maximum hailstone sequence length between a and b .

Example

Input:

2

1 10

20 25

Output:

9

22