J: TNumbers

Source file name: `tnumbers.c`, `tnumbers.cpp`, `tnumbers.java`, *or* `tnumbers.py`
Author: Rodrigo Cardoso

Note that

$$1 + 2 = 3,$$

but

$$\begin{aligned} 1 &\neq 2 + 3 + 4 \\ 1 + 2 &\neq 3 + 4 \\ 1 + 2 + 3 &\neq 4. \end{aligned}$$

A *TNumber* n is a positive integer for which there exist a number k, $1 \leq k < n$, such that the sum of the first k numbers equals the sum of the numbers from $k + 1$ to n. It is clear that 3 is a TNumber, but 4 is not.

Given two non-negative integers a and b, with $a \leq b$, determine how many numbers n satisfying $a \leq n \leq b$ are TNumbers.

Input

The problem input consists of several cases. A case is described with a line with two integer numbers a and b, $1 \leq a \leq b \leq 10^8$. The end of the input is signaled by a line with two zero values '`0 0`', which should not be processed.

The input must be read from standard input.

Output

For each case, output a line with exactly one integer value indicating how many TNumbers are there.

The output must be written to standard output.

Sample Input	Sample Output
`1 5`	`1`
`3 3`	`1`
`4 8`	`0`
`0 0`	