E. Periodical Numbers

time limit per test: 2 seconds

memory limit per test: 256 megabytes

input: standard input

output: standard output

A non-empty string s is called *binary*, if it consists only of characters "`0`" and "`1`". Let's number the characters of binary string s from 1 to the string's length and let's denote the i-th character in string s as s_i.

Binary string s with length n is *periodical*, if there is an integer $1 \le k < n$ such that:

- k is a divisor of number n
- for all $1 \le i \le n - k$, the following condition fulfills: $s_i = s_{i+k}$

For example, binary strings "101010" and "11" are periodical and "10" and "10010" are not.

A positive integer x is *periodical*, if its binary representation (without leading zeroes) is a periodic string.

Your task is to calculate, how many periodic numbers are in the interval from l to r (both ends are included).

Input

The single input line contains two integers l and r ($1 \le l \le r \le 10^{18}$). The numbers are separated by a space.

Please, do not use the `%lld` specifier to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams or the `%I64d` specifier.

Output

Print a single integer, showing how many periodic numbers are in the interval from l to r (both ends are included).

Examples

input

```
1 10
```

output

```
3
```

input

```
25 38
```

output

```
2
```

Note

In the first sample periodic numbers are 3, 7 and 10.

In the second sample periodic numbers are 31 and 36.