Deep Primes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	512 megabytes

For any non-negative integer x we can consider its decimal representation as a string of digits from 0 to 9. Denote this string as $S(x)$. On the opposite, for any string s of decimal digits we can get an integer it represents by neglecting all leading zeroes, except maybe one to represent integer 0. Denote this integer $D(s)$.

We say that integer y is an *integer-substring* of integer x if there exists a string s that is a substring (consecutive subsequence) of $S(x)$ and $D(s) = y$.

Recall that integer x is called prime if it is positive and has exactly two divisors. First five primes are $2, 3, 5, 7, 11$.

We call integer x *deep prime* if it is prime and any y that is integer-substring of x is prime.

You are given two integers n and m, calculate the number of deep prime integers between n and m, inclusive.

Input

The only line of the input contains two integers n and m ($1 \le n \le m \le 10^{18}$).

Output

Print one integer — the number of deep prime integers x that satisfy $n \le x \le m$.

Example

standard input	standard output
1 11	4