Problem A. Circular Primes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	256 megabytes
Java Class Name:	`circularprimes`
Maximum Points Available:	15

Recall that a prime number is a positive integer greater than 1 which is divisible only by itself and 1. The first few prime numbers are 2, 3, 5, 7, 11,...

Suppose you have some prime number P. You decide to move the first digit of P to the end of P. If you can do this process infinitely many times and always have a prime number, then the original number, P, is known as a **circular prime**.

For example, 1193 is a circular prime, since 1193, 1931, 9311 and 3119 are all prime. (Notice that after 3119, 1193 appears again, so the process will repeat). However, 1187 is **not** a circular prime, since 7118 is divisible by 2 (even though 1187 itself is prime).

Your task is, given a positive integer N, determine the N-th circular prime number.

Input

Input consists of a single integer N $(1 \leq N \leq 40)$

Output

Output a single integer - the N-th circular prime number.

Examples

standard input	standard output
5	11
10	71
15	131