

Next Prime

Function **is_prime(n)** below checks if **n** is a prime number. You have to implement **next_prime(n)** and **next_twin_prime(n)** by analyzing **is_prime(n)** following the comment in each function.

```
def is_prime(n):
    # Check if n is a prime number
    if n <= 1:
        return False
    for k in range(2, int(n**0.5)+1):
        if n%k == 0:
            return False
    return True

def next_prime(N):
    # Return the least prime number which is more than N

def next_twin_prime(N):
    # Return the least twin prime number which is more than N
    # twin prime number are 2 prime numbers differed by 2. For example -> (11 and 13) or (41 and 43).

exec(input().strip()) # This command is necessary to grade your answer
```

Input

Python commands used to test the functions.

Output

Result from executing the commands.

Example

Input (from keyboard)	Output (on screen)
<code>print(next_prime(1))</code>	2
<code>print(next_prime(20))</code>	23
<code>print(next_prime(10000000))</code>	10000019
<code>print(next_twin_prime(30))</code>	(41, 43)
<code>print(next_twin_prime(10000000))</code>	(10000139, 10000141)