

## Next Prime

ฟังก์ชัน `is_prime(n)` ข้างล่างนี้ตรวจสอบว่า `n` ที่ได้รับเป็นจำนวนเฉพาะหรือไม่ จงใช้ `is_prime` ให้เป็นประโยชน์ในการเขียนฟังก์ชัน

`next_prime(n)` และ `next_twin_prime(n)` ตามคำอธิบายที่เขียนใน comment ของแต่ละฟังก์ชัน

```
def is_prime(n):
    # ทดสอบว่า n เป็นจำนวนเฉพาะหรือไม่
    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):
    # คืนจำนวนเฉพาะตัวที่มีค่าน้อยสุดที่มากกว่า N

def next_twin_prime(N):
    # คืนจำนวนเฉพาะสองค่าที่เป็น twin prime ที่มีค่าน้อยสุดที่มากกว่า N
    # twin prime คือจำนวนเฉพาะที่มีค่าต่างกัน 2 เช่น 11 กับ 13 หรือ 41 กับ 43

exec(input().strip()) # ต้องมีคำสั่งนี้ ตรงนี้ ดอนส่งให้ Grader ตรวจ
```

## ข้อมูลนำเข้า

คำสั่งภาษา Python ที่ใช้ทดสอบการทำงานของฟังก์ชัน

## ข้อมูลส่งออก

ผลที่ได้จากการสั่งทำงานคำสั่งที่ได้รับ

## ตัวอย่าง

input (จากแป้นพิมพ์)	output (ทางจอภาพ)
<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)