```
from unittest import result
from sqlalchemy import false, true
from sympy import pprint, re
from sympy import primefactors, factorint

class NumberTheory:
    def __init__(self,interger,coso):
        self.interger = interger
        self.coso = coso

    def getInt(self):
        return self.interger

    def check_prime_number(self):
        flag = 1
        if (self.interger <2):
            flag = 0
            return flag

        for p in range(2, self.interger):
            if self.interger % p == 0:
                flag = 0
                break
        return flag

    def kiemtrasogianguyento(self):
        result = ((self.coso**self.interger)-self.coso)%self.interger
        if(result == 0):
            print("%s la so gia nguyen to" % (self.interger))
        else:
            print("%s khong la so gia nguyen to" % (self.interger))
    def check_Carmichael(self):
        arr_prime = primefactors(self.interger)

        if(len(arr_prime)>=3):

            result1 = (self.interger - 1)%(arr_prime[0]-1)

            result2 = (self.interger - 1)%(arr_prime[1]-1)

            result3 = (self.interger - 1)%(arr_prime[2]-1)

            if(result1 == result2 == result3 == 0):
                print("%s is Carmichael" % self.interger)
        else:
```

```
            print("%s is not Carmichael" % self.interger)

a = int(input("So nguyen duong: "))
n = int(input("Nhap co so: "))
# check = isinstance(a,int)
while(a <= 0):
    print("Input again\n")
    a = int(input("So nguyen duong: "))

number = NumberTheory(a,n)

# print(number.interger)
if(number.check_prime_number()==1):
    print("%s la so nguyen to\n" % (number.interger))
else:
    print("%s la so hop to\n" % (number.interger))

print(factorint(a))

number.kiemtrasogianguyento()

number.check_Carmichael()
```

```
So nguyen duong: 561
Nhap co so: 8
561 la so hop to

{3: 1, 11: 1, 17: 1}
561 la so gia nguyen to
561 is Carmichael
```

```
So nguyen duong: 65
Nhap co so: 8
65 la so hop to

{5: 1, 13: 1}
65 la so gia nguyen to
65 is not Carmichael
```