

Part1:

Câu 1:

- #hàm kiểm tra số nguyên tố

```
def check_prime_number(n):
```

```
    flag = 1
    if (n < 2):
        flag = 0
        return flag
    for p in range(2, n):
        if n % p == 0:
            flag = 0
            break
    return flag
```

```
check_prime_number(11)
```

```
def check_min_2_prime(a):
```

```
    count=0
    for i in range(0, len(a)):
        if (check_prime_number(a[i]) == 1): # nếu a[i] là số nguyên tố
            count=count+1 # tăng biến đếm
    if count >= 2:
        return True
    else :
        return False
```

#mảng không có số nguyên tố

```
a=[1,4,6,8]
res=check_min_2_prime(a)
res
```

#mảng có 1 số nguyên tố

```
a=[1,2,4,6]
res1=check_min_2_prime(a)
res1
```

#mảng nhiều hơn 2 số nguyên tố

```
a=[1,2,3,5]
res2=check_min_2_prime(a)
res2
```

✓
0
giây

▶

```
res

#mảng có 1 số nguyên tố
a=[1,2,4,6]
res1=check_min_2_prime(a)
res1

#mảng nhiều hơn 2 số nguyên tố
a=[1,2,3,5]
res2=check_min_2_prime(a)
res2
```

True

Q

✓

[13] #hàm kiểm tra số nguyên tố

def check_prime_number(n):

flag = 1

if (n < 2):

flag = 0

return flag

for p in range(2, n):

if n % p == 0:

flag = 0

break #Chỉ cần tìm thấy 1 ước số là đủ và thoát vòng lặp

return flag

✓

[11] check_prime_number(11)

0
giây

1

✓

▶

```
def check_min_2_prime(a):
    count=0
    for i in range(0,len(a)):
        if(check_prime_number(a[i])==1): # nếu a[i] là số nguyên tố
            count=count+1 # tăng biến đếm
    if count>=2:
```

↑

↓

↺

⚙

📄

🗑

⋮

0 giây

hoàn thành lúc 15:45

```
[18] #1b mảng không có số nguyên tố
a=[1,4,6,8]
res=check_min_2_prime(a)
res

False

[15] #mảng có 1 số nguyên tố
a=[1,2,4,6]
res1=check_min_2_prime(a)
res1

False

[16] #mảng nhiều hơn 2 số nguyên tố
a=[1,2,3,5]
res2=check_min_2_prime(a)
res2

True

[19] #mảng là chuỗi
# câu 2:
import math

def module(self):
    return abs(self.real_number_)

check=sothuc()
#check.real_number_=-17
class sophuc(sothuc):
    image_number=3
    def __init__(self):
        sothuc.__init__(self)
        print(" tạo class so phuc")
    def module(self):
        print(" hàm module")
        return sqrt(self.image_number**2+self.real_number_**2)

i=sophuc()
res=i.module()
res

done
done
tạo class so phuc
4
```

Part 2:

Câu 2:

a)

SELECT SUM(commission), city FROM salesman GROUP BY city;

b)

select * from customer c

join salesman s on c.salesman_id = s.salesman_id

where s.commission > 0.12

order by s.commission