**Phần 1:**

Câu 1a:

def KT(arr):

count =0

for i in arr:

if(i<2):

count=count+1

else:

for j in range (2,i-1):

if(i%j==0 and j<i):

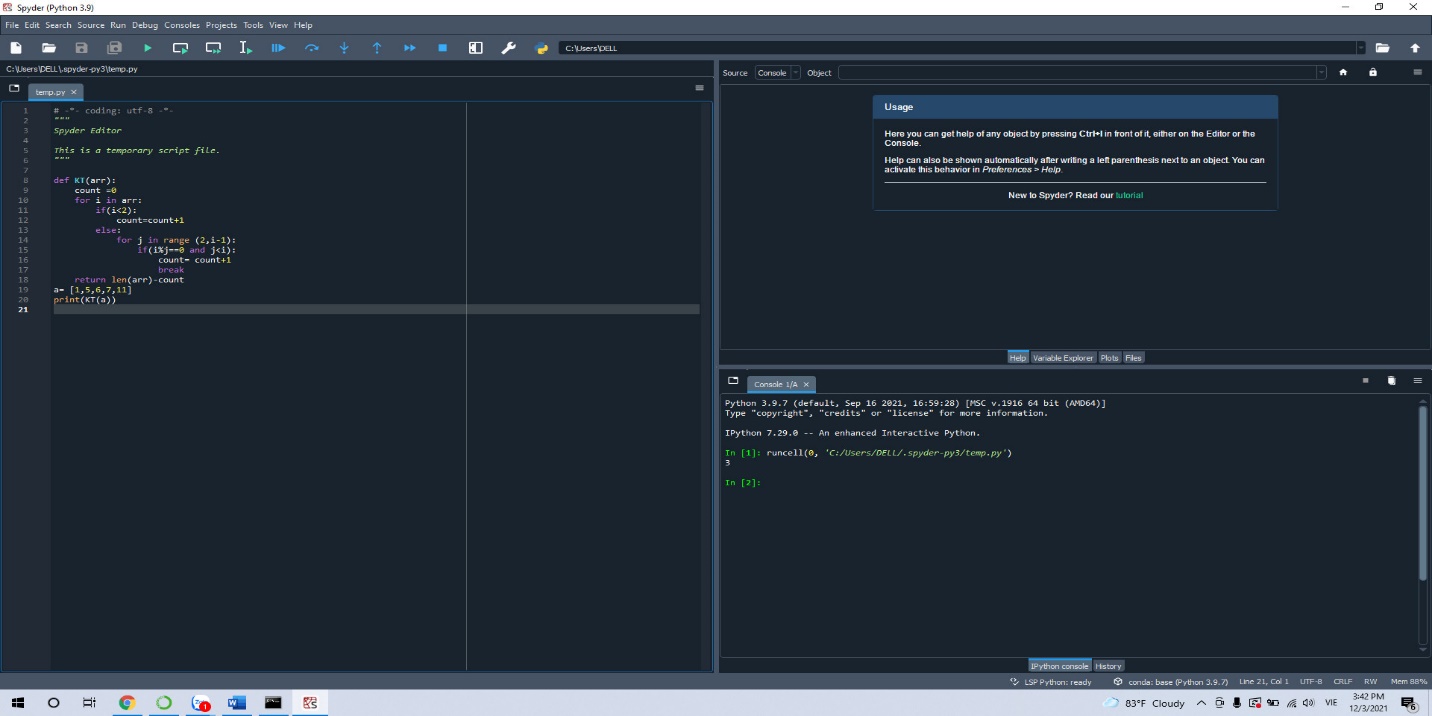
count= count+1

break

return len(arr)-count

a= [1,5,6,7,11]

print(KT(a))



Câu 1b:

#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 có phần tử không phải số nguyên

a=[2,4,6,8]

res2=check\_min\_2\_prime(a)

res2

**Phần 2:**

Câu 1:

nput("Moi nhap real\_number: "))

m = float(input("Moi nhap image\_number: "))

class Sothuc :

def \_\_init\_\_(self,real\_number) :

self.real\_number = real\_number

def gttd(self) :

return (self.real\_number \*2) \* 0.5

class Sophuc(Sothuc) :

def \_\_init\_\_(self,real\_number, image\_number) :

super().\_\_init\_\_(real\_number)

self.image\_number = image\_number

def module(self) :

return (self.image\_number\*\*2 + self.real\_number\*\*2) \*\* 0.5

sothuc = Sothuc(n)

print("Tri tuyet doi co gia tri la: ",sothuc.gttd())

sophuc = Sophuc(n,m)

print("module cua so phuc la: ",sophuc.module())