TRƯỜNG ĐẠI HỌC VINH

# VIỆN KỸ THUẬT CÔNG NGHỆ



# BÁO CÁO THỰC HÀNH

**KỸ THUẬT LẬP TRÌNH**

MÃ HỌC PHẦN: ELE20004

SVTH: NGUYỄN VIẾT TRUNG MSSV: 245752021610157

## GVHD: MAI THẾ ANH

**NGHỆ AN - 2025**

**MỤC LỤC**

**Bài 1**. Thực hiện các thuật toán bằng phần mềm Flowgorithm .....................................2

**Bài 2**. Các cú pháp, kiểu dữ liệu, lệnh điều khiển trong lập trình Python ......................4

**Bài 3**. Lập trình hàm trong Python…............................................................................14

**Bài 4**. Các kiểu dữ liệu có cấu trúc trong Python..........................................................21

**Bài 5**. Thiết kế module trong Python..5........................................................................34

**Bài 6:** Lập trình hướng đối tượng trong Python............................................................40

**Bài 7:** Thao tác trên tập tin và thư mục trong Python...................................................44

**Bài 8:** Lập trình giao diện trong Python........................................................................47

**Bài 1. Thực hiện các thuật toán bằng phần mềm Flowgorithm**

### 1.1. Mục đích

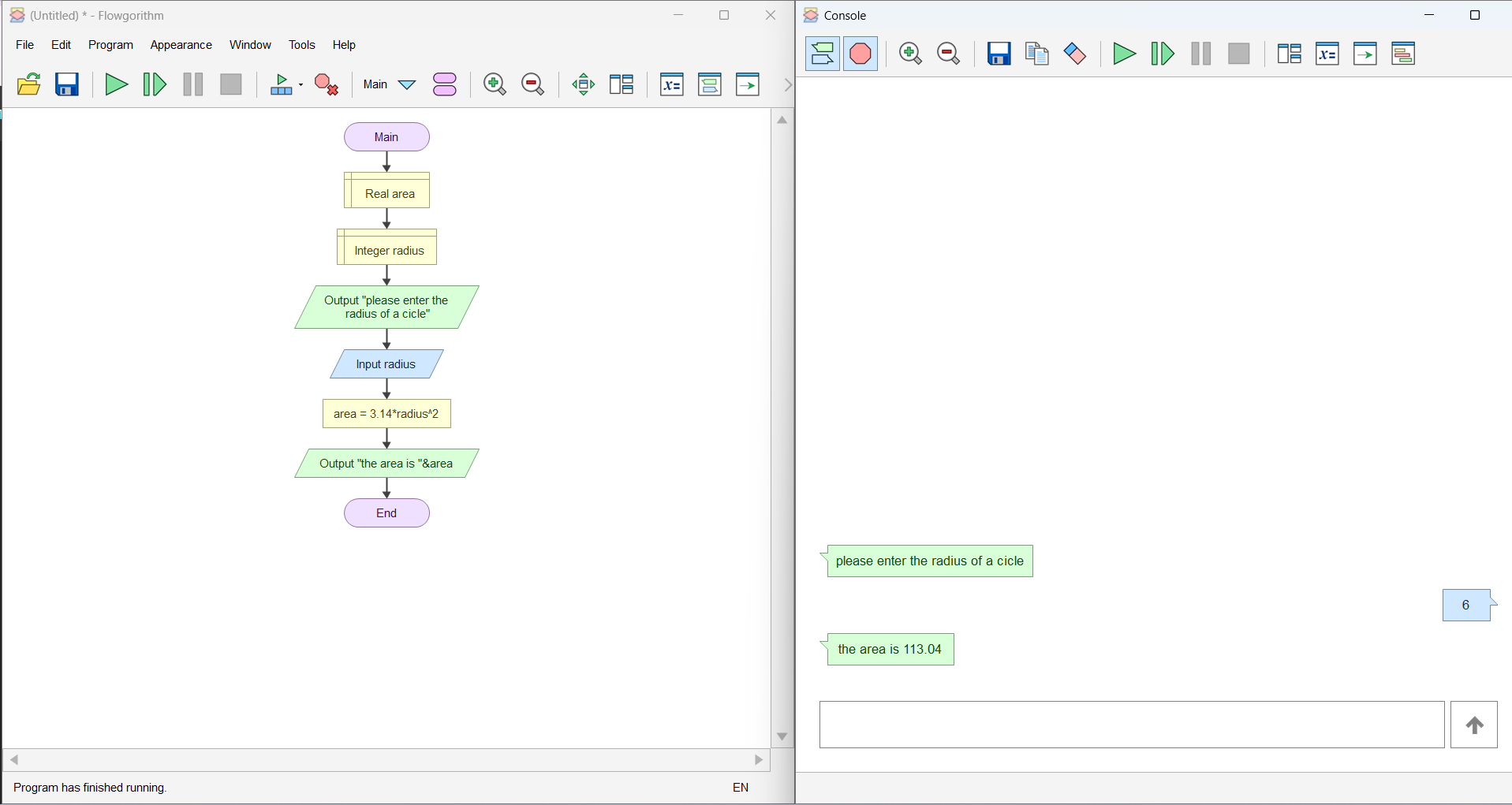
* Sử dụng phần mềm Flowgorithm trong thiết kế và biểu diễn thuật toán;
* Xây dựng thuật toán cho các bài toán cụ thể trên Flowgorithm

a. Tính diện tích hình tròn

**Các bước thực hiện:**

1. Thêm khối Input để nhập biến r (bán kính).
2. Thêm khối Assignment: S = 3.14 \* r \* r.
3. Thêm khối Output để in S.
4. Chạy chương trình và nhập giá trị r.
5. **Kết quả nhận được**

* Với r = 2 → in Diện tích = 12.56.



b. Nhập và in số lượng chai sử dụng vòng lặp for

**Các bước thực hiện**

1. Thêm khối Input để nhập n (số lượng chai).
2. Dùng khối Loop (For or While) từ i = 1 đến n.
3. Trong vòng lặp thêm khối Output in i hoặc thông tin tương ứng.
4. Chạy chương trình và thử với n = 5.

**Kết quả nhận được**

* In ra dãy: 1, 2, 3, 4, 5 (mỗi giá trị trên 1 dòng).

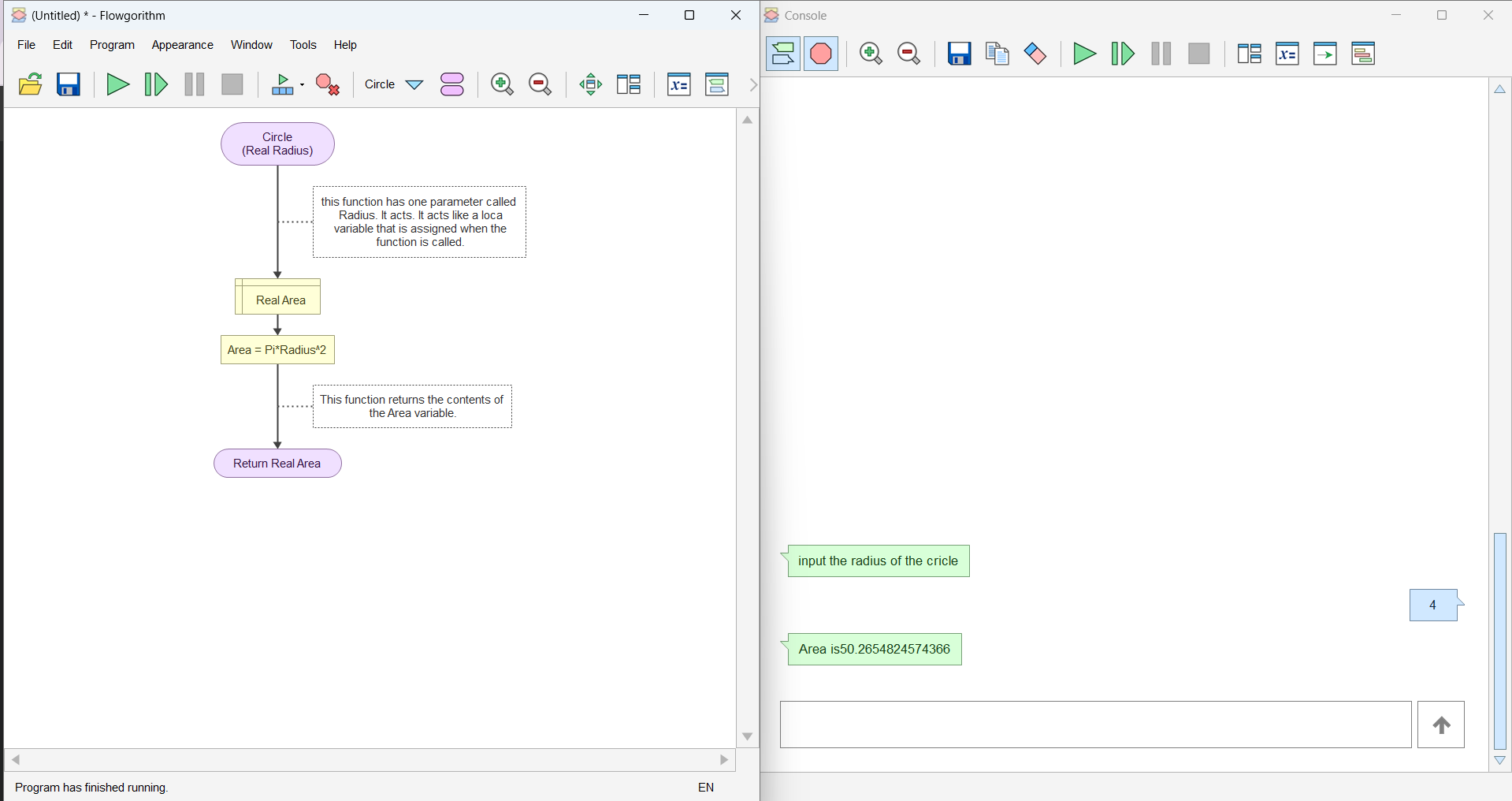
c. Tính diện tích hình tròn sử dụng hàm (chương trình con)

**Các bước thực hiện**

1. Tạo subprogram TinhDienTich(r) trả về 3.14 \* r \* r.
2. Trong Main: Input r → gọi TinhDienTich(r) → gán S → Output S.
3. Chạy chương trình để kiểm tra.

**Kết quả nhận được**

* Kết quả giống Bài 2a; hàm trả về và Main in diện tích.



**1.3. Ý nghĩa các khối sử dụng trong chương trình**

| Tên khối | Ý nghĩa – Chức năng |
| --- | --- |
| Start / End (Terminal) | Là điểm bắt đầu và kết thúc của chương trình. Mọi lưu đồ đều bắt đầu từ *Start (Main)* và kết thúc ở *End*. | |
| Input / Output (Dữ liệu vào/ra) | - Input: dùng để nhập dữ liệu từ bàn phím (ví dụ: nhập r, n, a, h …). - Output: dùng để hiển thị kết quả hoặc thông báo ra màn hình (ví dụ: in diện tích, in thông tin SV). | |
| Assignment (Gán / Tính toán) | Dùng để thực hiện phép gán hoặc tính toán, như S = 3.14 \* r \* r hoặc i = i + 1. | |
| If / Selection (Rẽ nhánh) | Dùng để kiểm tra điều kiện và rẽ nhánh theo hai hướng *True/False*. Ví dụ: kiểm tra n % 2 == 0 để xác định số chẵn/lẻ. | |
| Loop (Vòng lặp) | Dùng để lặp lại một đoạn chương trình nhiều lần, ví dụ for i = 1 to n. | |
| Function / Call (Hàm con) | Dùng để tạo và gọi chương trình con (subprogram), giúp tái sử dụng mã. Ví dụ: TinhDienTich(r). | |
| Comment (Chú thích) | Dùng để ghi chú giải thích chương trình, không ảnh hưởng đến quá trình thực thi. | |

1.4. Tài liệu thamkhảo

http://www.flowgorithm.org/documentation/index.htm

**Bài 2. Các cú pháp, kiểu dữ liệu, lệnh điều khiển trong lập trình Python**

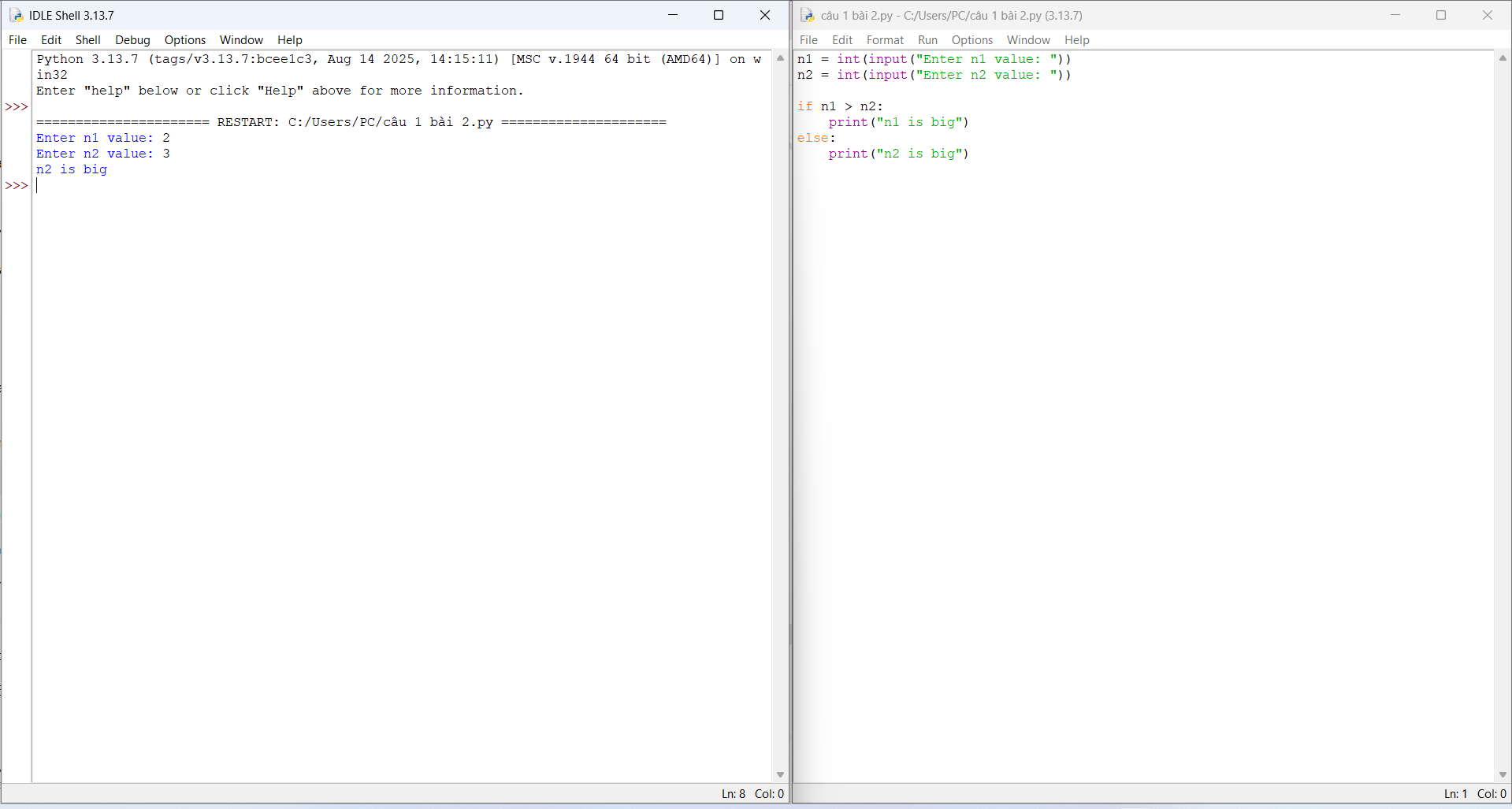
**2.1. Mục đích**

### Giúp sinh viên nắm bắt các kiến thức cơ bản như kiểu dữ liệu, các kiểu cú pháp, các lệnh điều khiển khi viết một chương trình bằng ngôn ngữ python.

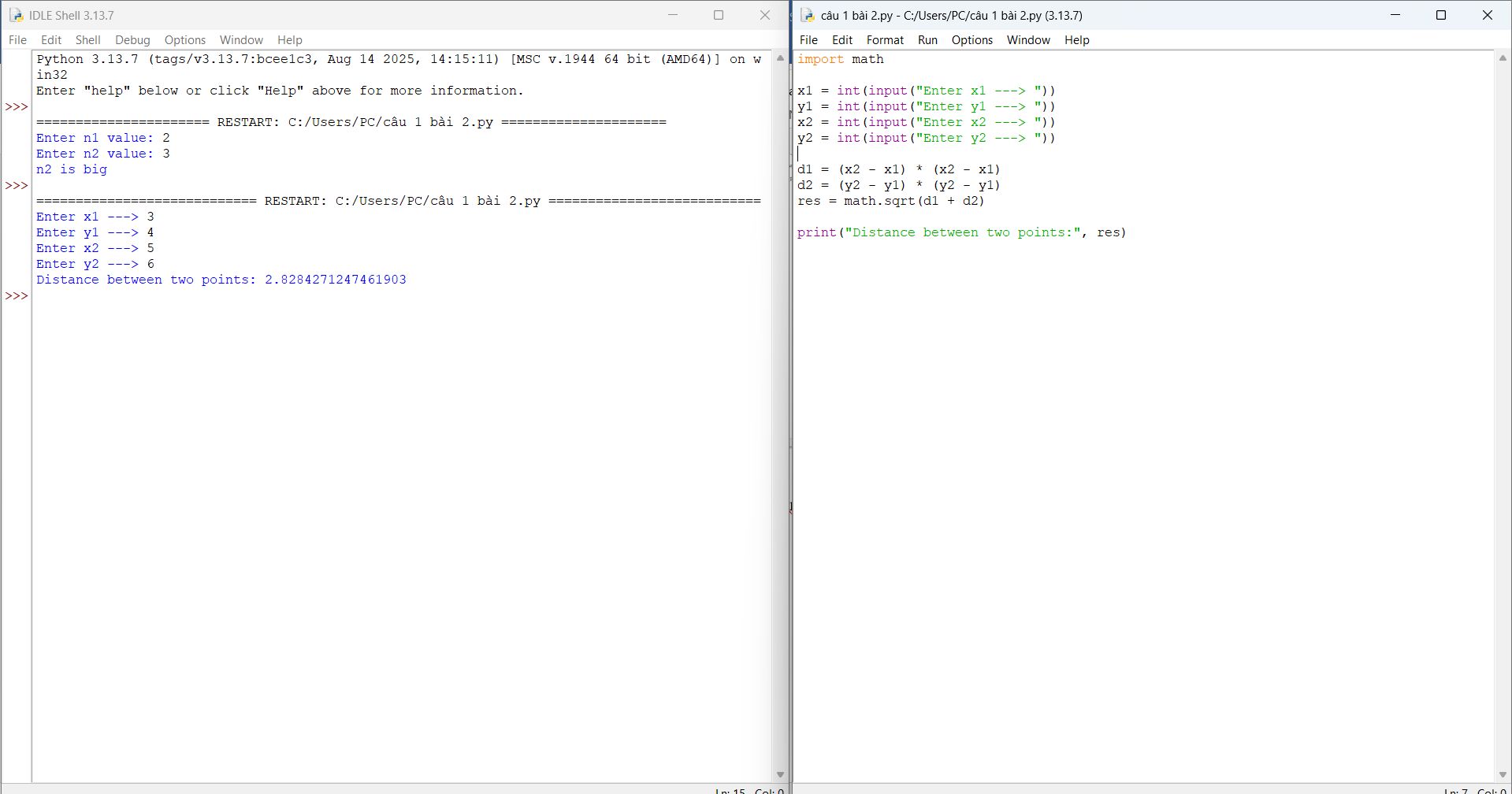
### 2.2. Các bước thực hiện và kết quả

1. Viết đoạn chương trình sau và sửa lỗi.

-Đoạn chương trình trên bị lỗi cú pháp được sửa lại và kết quả như sau:

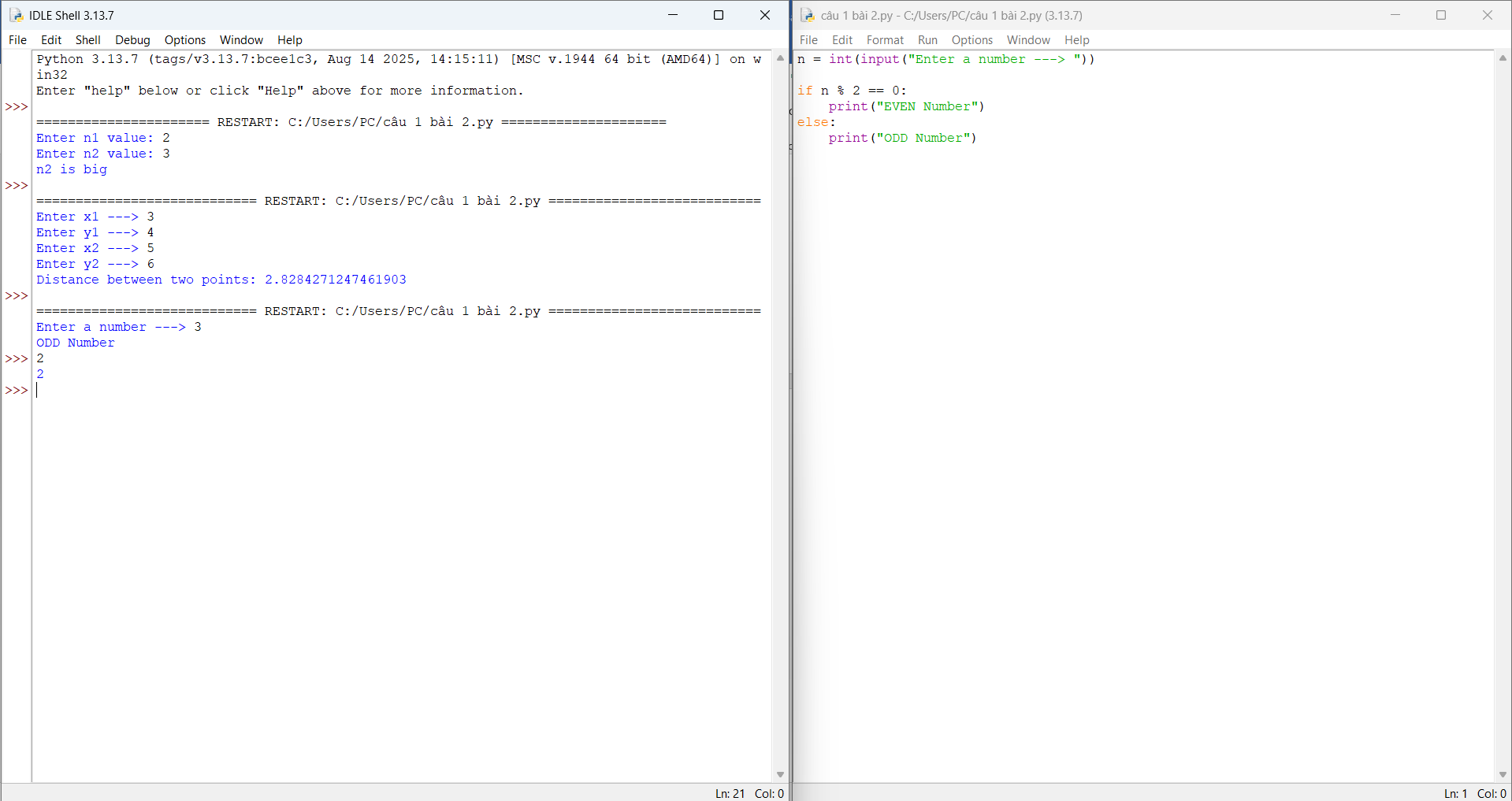


1. Viết chương trình nhập hai điểm và tính khoảng cách.

****

3. Viết chương trình nhập vào một số và kiểm tra số đó là chẵn hay lẻ, in thông báo ra màn hình.

-kết quả cho ra:

****

4. Viết chương trình in ra màn hình số nghịch đảo và kết quả dưới dạng thập phân của một dãy số tự nhiên trong khoảng (a,b).

-Chương trình và ví dụ như sau:

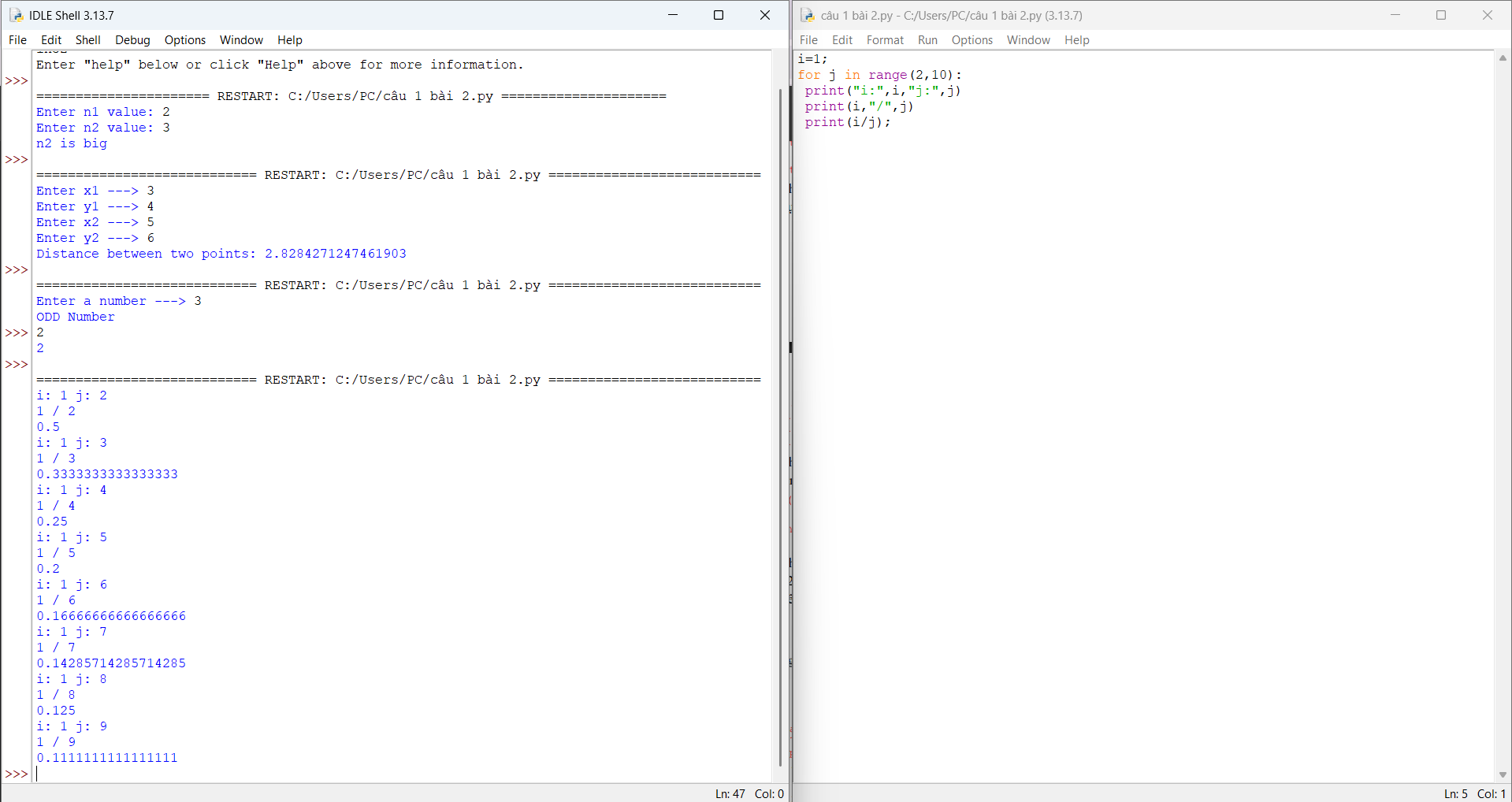
i=1;

for j in range(2,10):

print("i:",i,"j:",j)

print(i,"/",j)

print(i/j);



5. Viết chương trình nhập vào một số tự nhiên n > 0, in ra màn hình các số tự nhiên giảm dần từ n đến 0, mỗi ký tự in trên 1 hàng

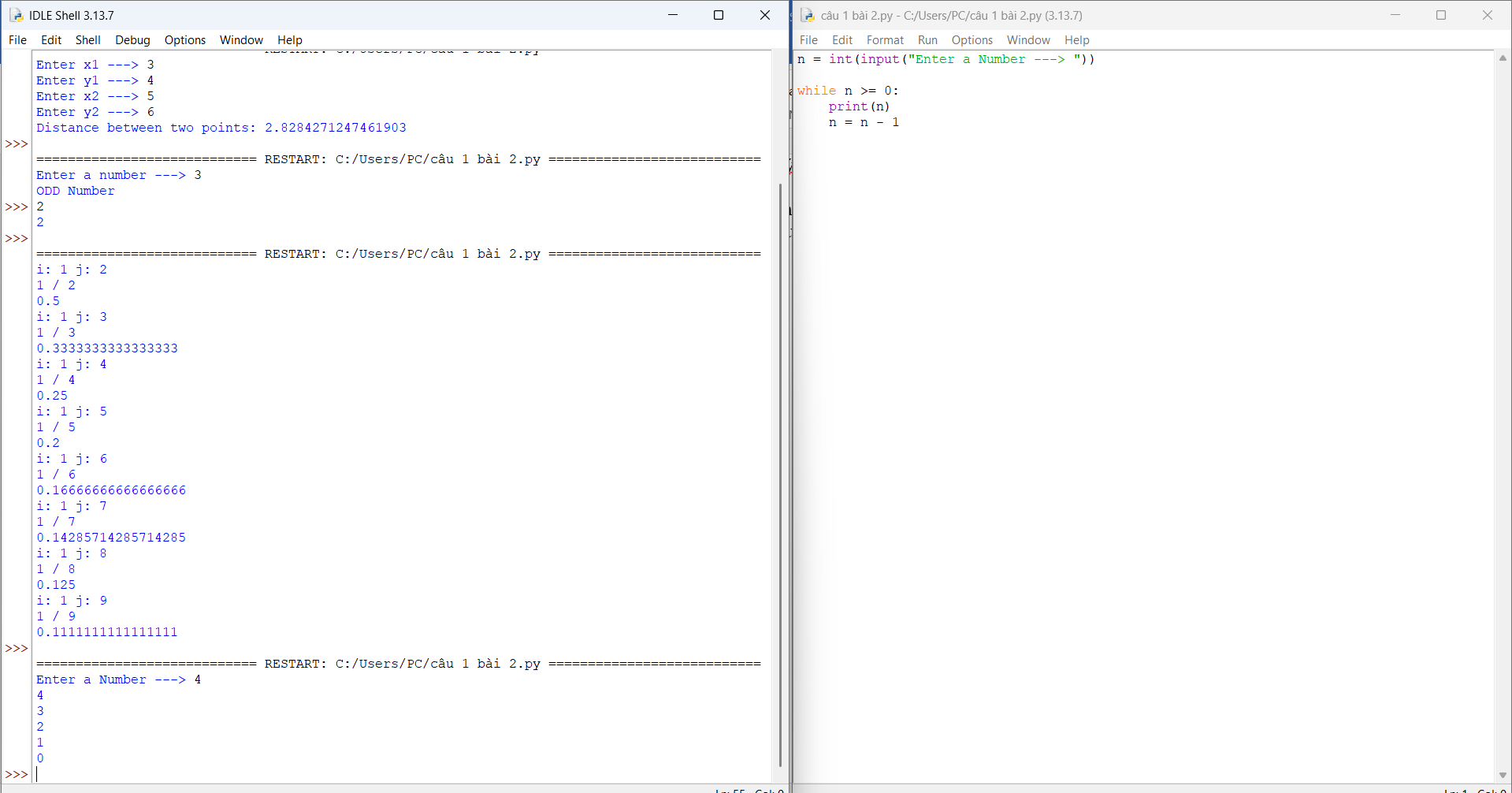
-Chương trình và ví dụ như sau:

n = int(input("Enter a Number ---> "))

while n >= 0:

print(n)

n = n - 1

****

6. Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội số của 5, nằm trong đoạn 2000 và 3200 (tính cả 2000 và 3200). Các số thu được sẽ được in thành chuỗi trên một dòng, cách nhau bằng dấu phẩy.

-Chương trình và ví dụ như sau:

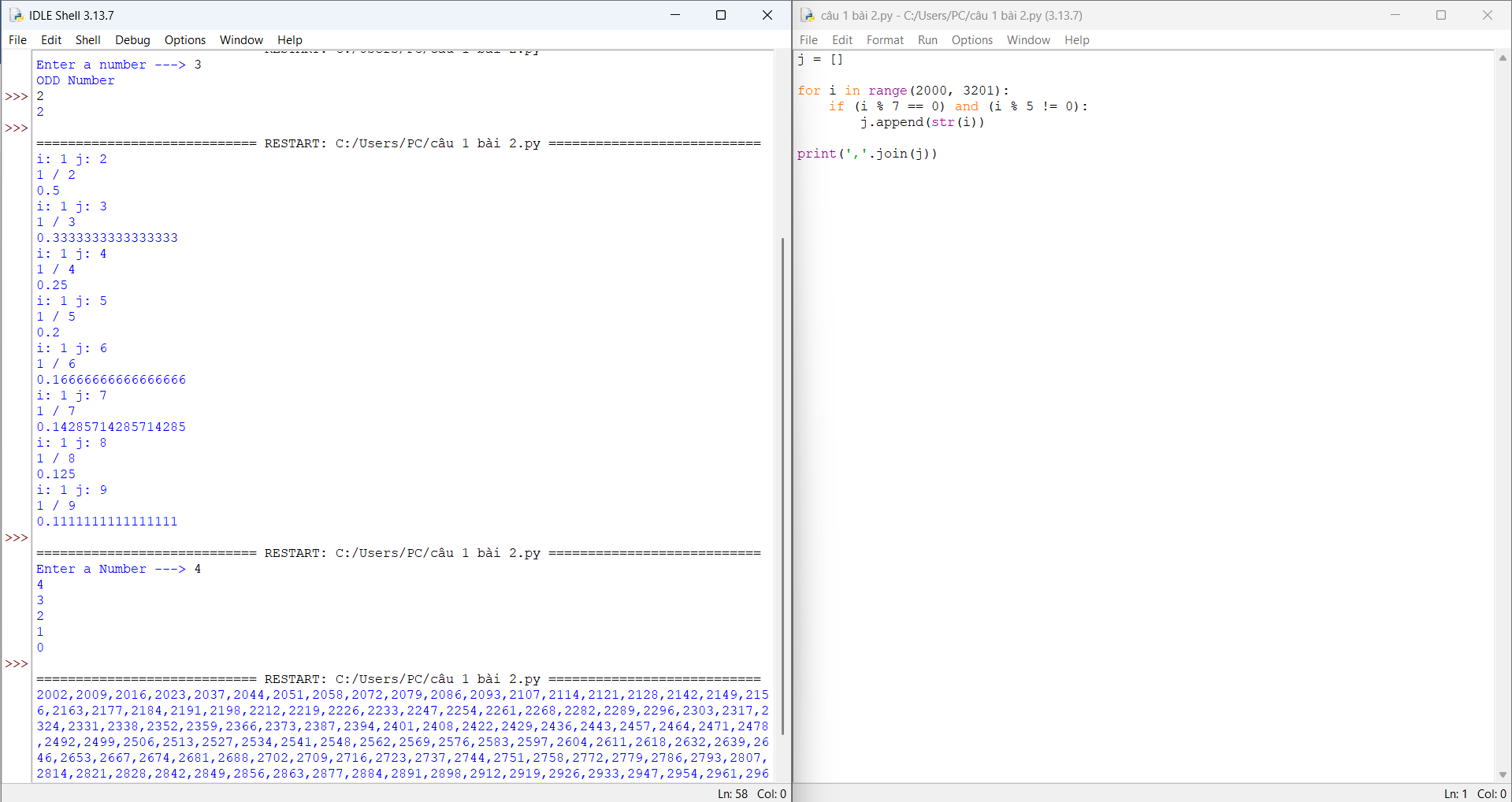
j = []

for i in range(2000, 3201):

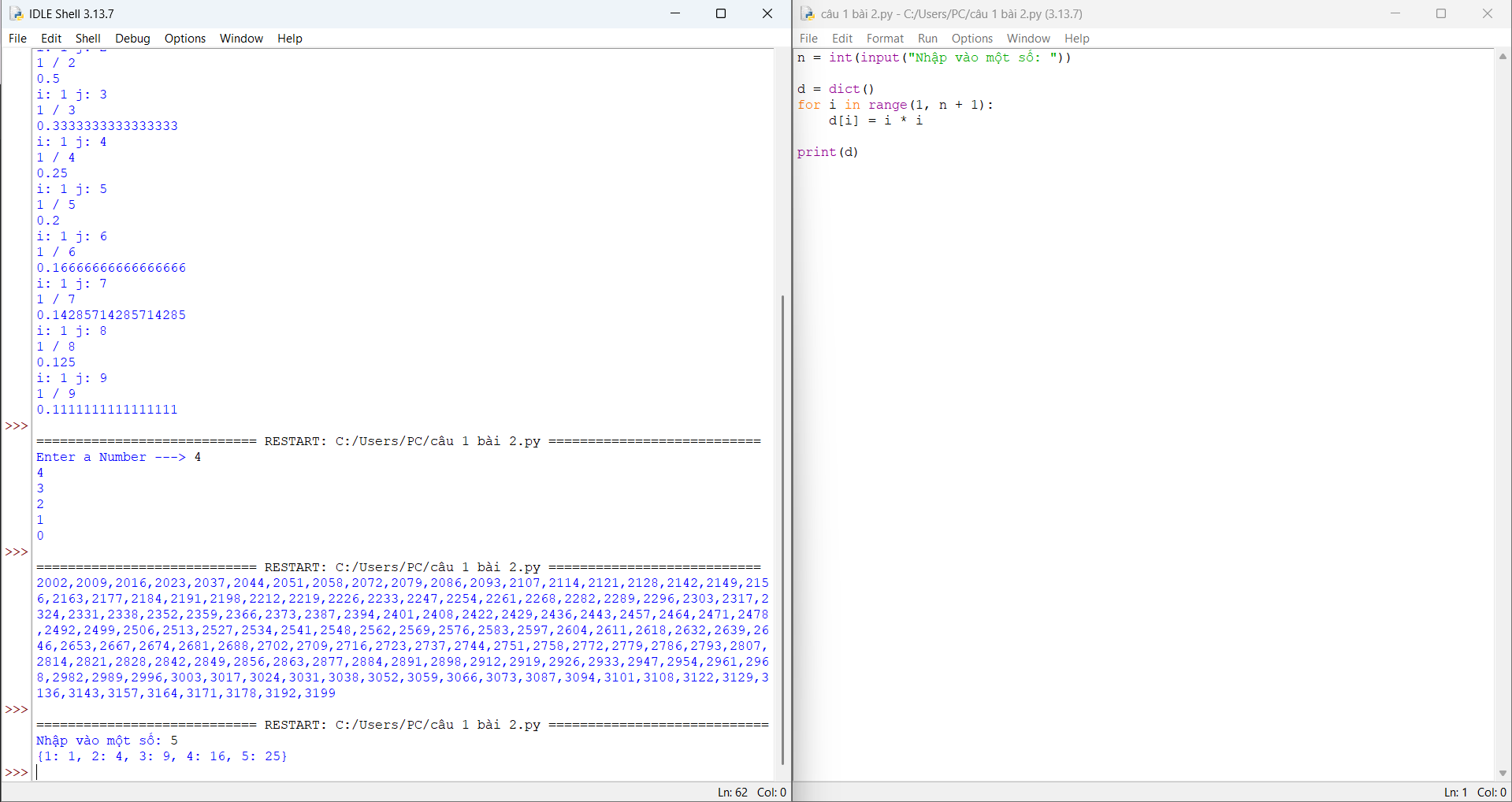
if (i % 7 == 0) and (i % 5 != 0):

j.append(str(i))

print(','.join(j))



7. Với số nguyên n nhất định, hãy viết chương trình để tạo ra một dictionary chứa (i, i\*i) như là số nguyên từ 1 đến n (bao gồm cả 1 và n) sau đó in ra dictionary này.

****

8. Viết chương trình in ra màn hình dãy số Fibonacci nhỏ hơn 4.000.000, tìm tổng các số chẵn trong dãy đã in

-Chương trình và tổng các sỗ chẵn trong dãy in là:

a, b = 1, 2

total = 0

print(a, end=" ")

print(b, end=" ")

while a < 4000000:

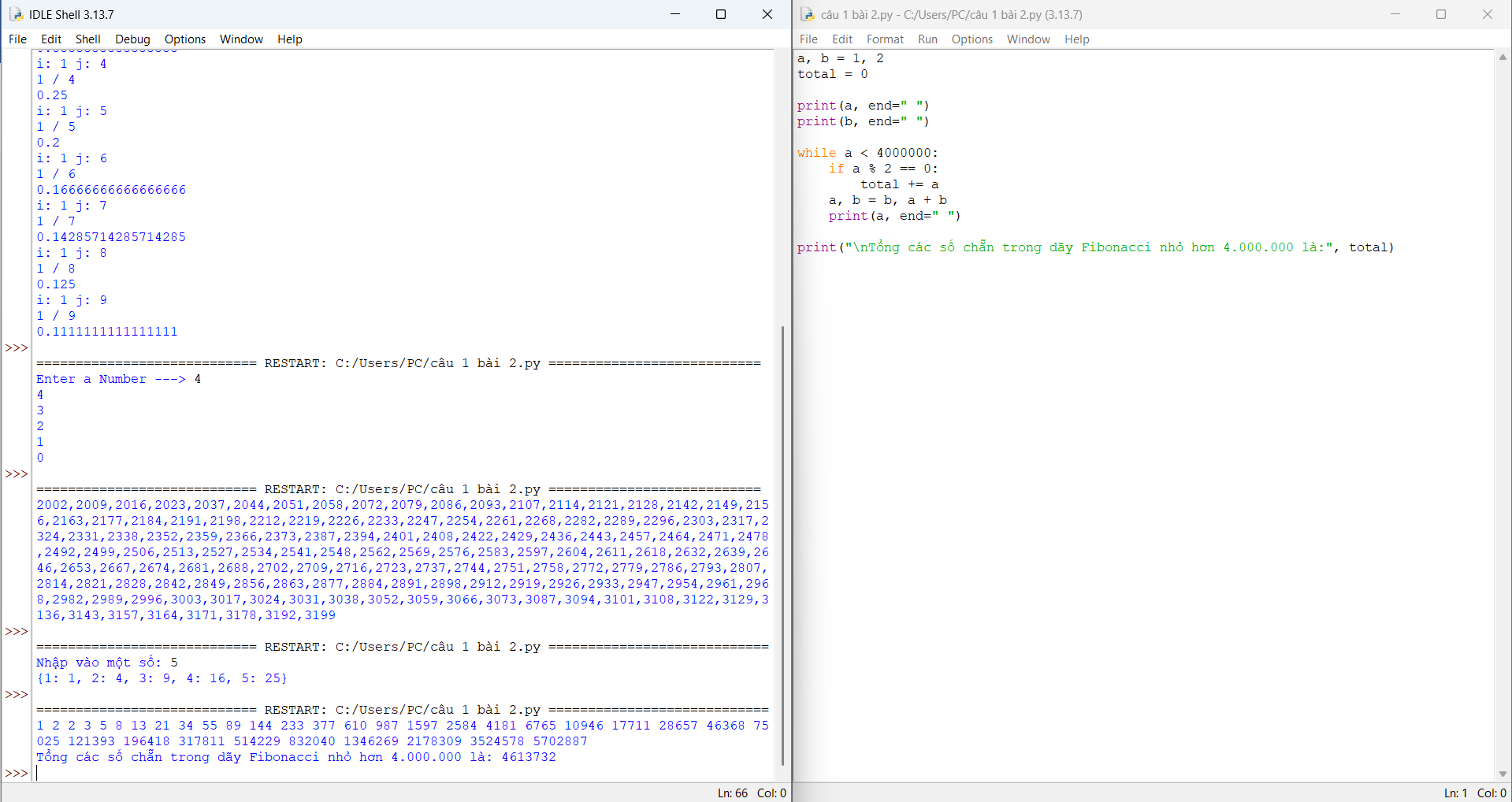
if a % 2 == 0:

total += a

a, b = b, a + b

print(a, end=" ")

print("\nTổng các số chẵn trong dãy Fibonacci nhỏ hơn 4.000.000 là:", total)

****

9. Viết chương trình đếm số ký tự trong 1 xâu ký tự nhập vào từ bàn phím, lưu các ký tự vào cấu trúc từ điển

-Chương trình như sau:

s = input("Enter a String: ")

d = {}

for i in s:

if i in d:

d[i] += 1

else:

d[i] = 1

print(d)



10.Viết chương trình sử dụng các phương thức split và join để tách nhập xâu ký tự

-Chương trình như sau:

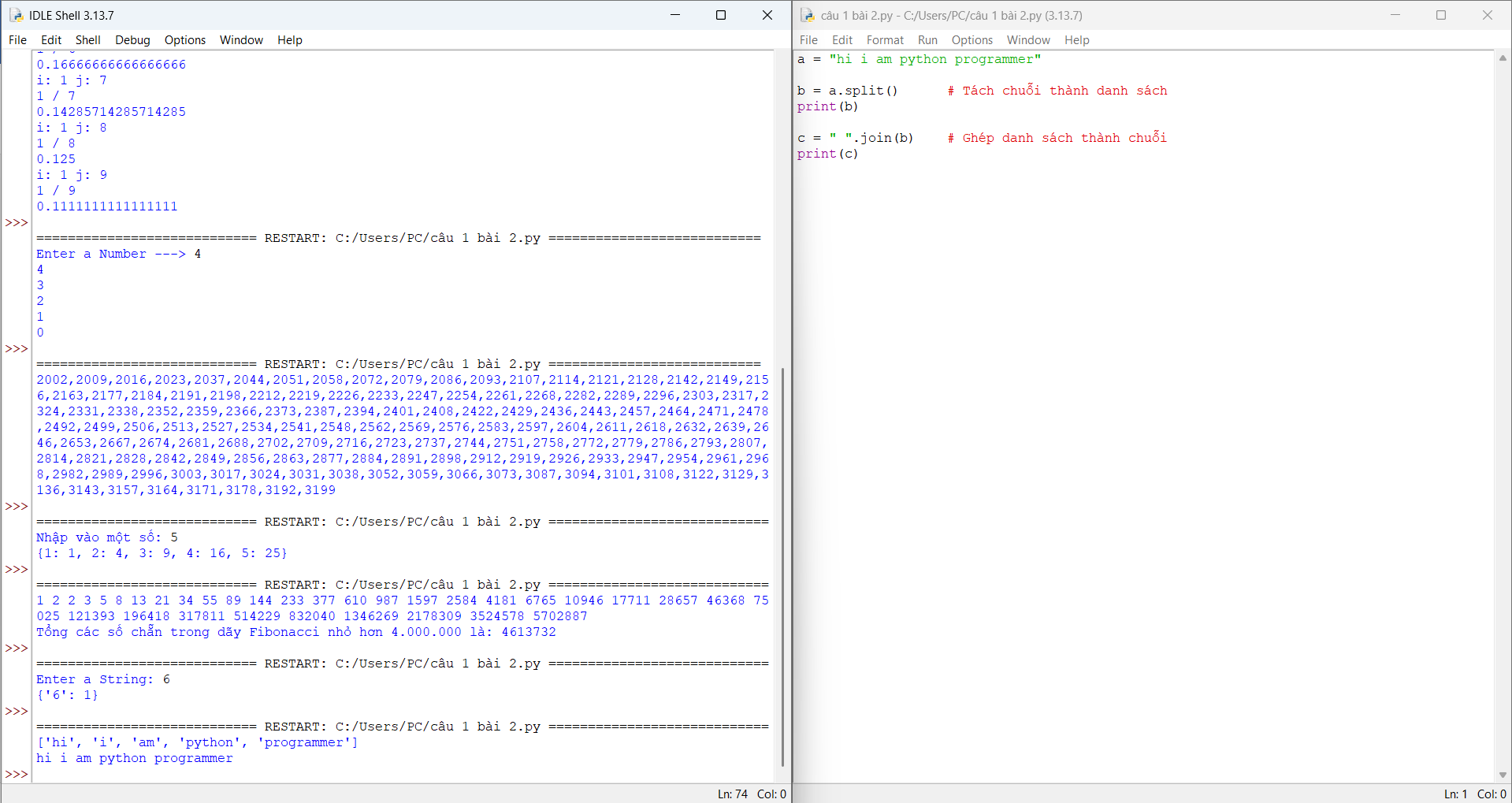
a = "hi i am python programmer"

b = a.split() # Tách chuỗi thành danh sách

print(b)

c = " ".join(b) # Ghép danh sách thành chuỗi

print(c)

****

11.Viết chương trình kết nối các danh sách vào từ điển

-Chương trình như sau:

l = [1, 'python', 4, 7]

k = ['cse', 2, 'guntur', 8]

m = []

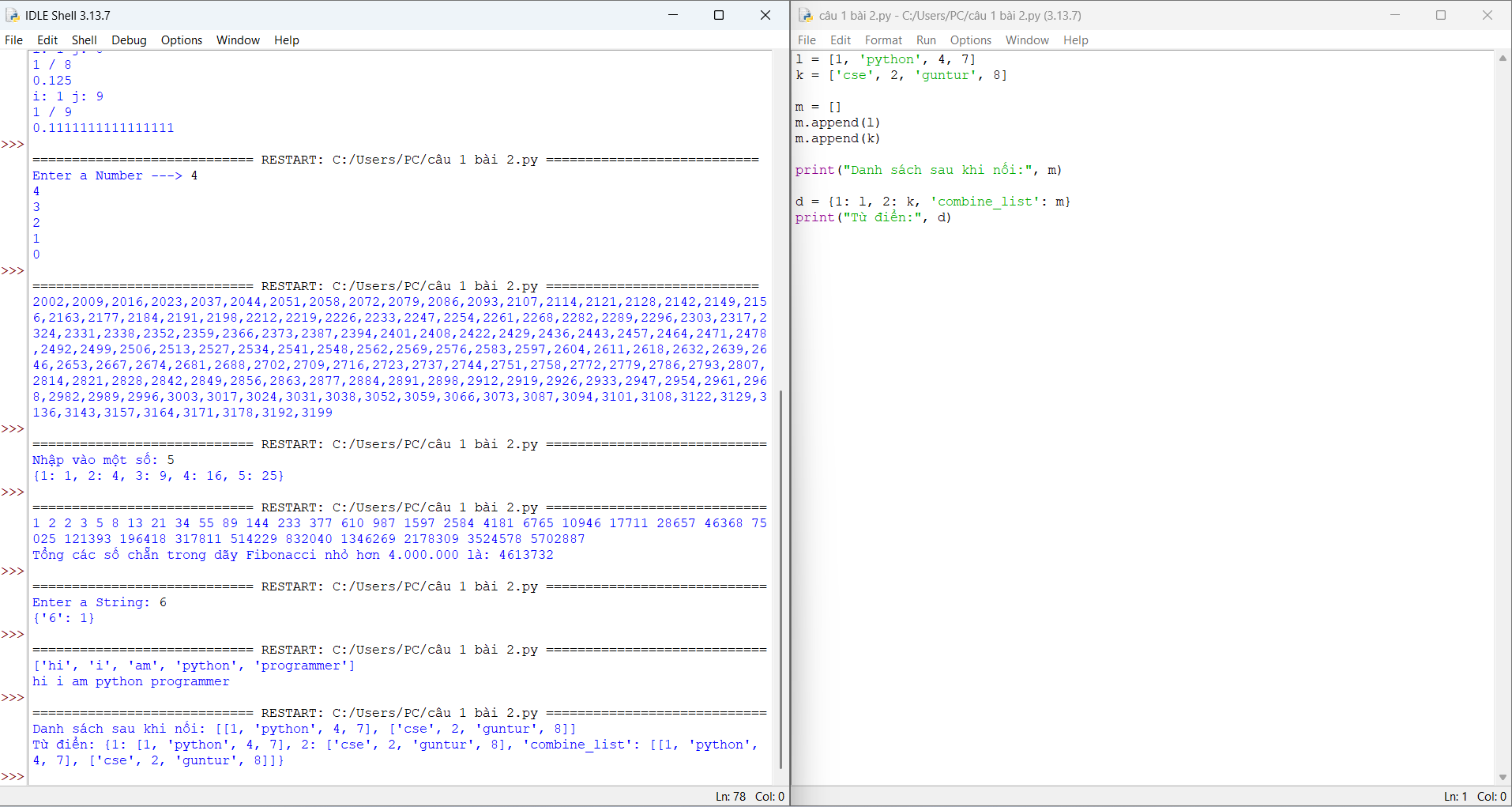
m.append(l)

m.append(k)

print("Danh sách sau khi nối:", m)

d = {1: l, 2: k, 'combine\_list': m}

print("Từ điển:", d)



12.Một website yêu cầu người dùng nhập tên người dùng và mật khẩu để đăng ký. Viết chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào. Các tiêu chí kiểm tra mật khẩu bao gồm: - Ít nhất 1 chữ cái nằm trong [a-z] - Ít nhất 1 số nằm trong [0-9] - Ít nhất 1 kí tự nằm trong [A-Z] - Ít nhất 1 ký tự nằm trong [$ # @] - Độ dài mật khẩu tối thiểu: 6 - Độ dài mật khẩu tối đa: 12 Chương trình phải chấp nhận một chuỗi mật khẩu phân tách nhau bởi dấu phẩy và kiểm tra xem chúng có đáp ứng những tiêu chí trên hay không. Mật khẩu hợp lệ sẽ được in, mỗi mật khẩu cách nhau bởi dấu phẩy

-Chương trình như sau:

import re

value = []

items = [x for x in input("Nhập các mật khẩu (phân cách bởi dấu phẩy): ").split(',')]

for p in items:

if len(p) < 6 or len(p) > 12:

continue

if not re.search("[a-z]", p):

continue

if not re.search("[0-9]", p):

continue

if not re.search("[A-Z]", p):

continue

if not re.search("[$#@]", p):

continue

value.append(p)

print("Các mật khẩu hợp lệ là:", ",".join(value))

13.Viết chương trình giải phương trình bậc 2: ax2 +bx+c=0, với các hệ số a, b, c nhập từ bàn phím

Chương trình như sau:

import math

print("Giải phương trình bậc hai: ax^2 + bx + c = 0")

a = float(input("Nhập a: "))

b = float(input("Nhập b: "))

c = float(input("Nhập c: "))

if a == 0:

# Trường hợp phương trình bậc nhất: bx + c = 0

if b == 0:

if c == 0:

print("Phương trình có vô số nghiệm.")

else:

print("Phương trình vô nghiệm.")

else:

x = -c / b

print("Phương trình có một nghiệm duy nhất: x =", x)

else:

# Phương trình bậc hai

delta = b\*\*2 - 4\*a\*c

if delta < 0:

print("Phương trình vô nghiệm thực.")

elif delta == 0:

x = -b / (2\*a)

print("Phương trình có nghiệm kép: x1 = x2 =", x)

else:

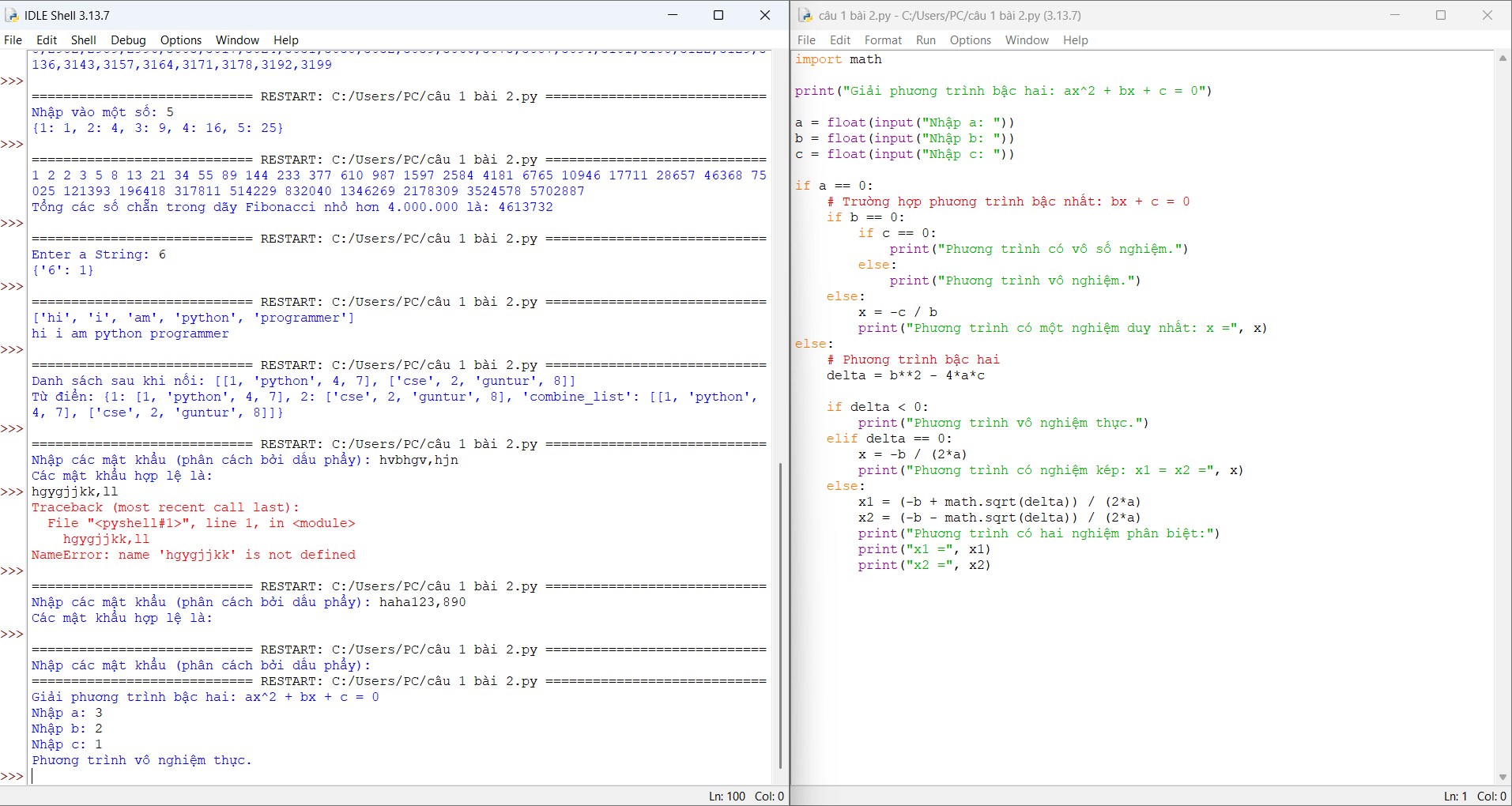
x1 = (-b + math.sqrt(delta)) / (2\*a)

x2 = (-b - math.sqrt(delta)) / (2\*a)

print("Phương trình có hai nghiệm phân biệt:")

print("x1 =", x1)

print("x2 =", x2)

****

**3.3. Câu hỏi kiểm tra**

### a. Các kiểu biến, khai báo và đặt tên biến trong Python

* Các kiểu dữ liệu: int, float, str, bool, list, tuple, set, dict
* Cách khai báo: a = 5, b = "text", c = [1,2,3]
* Quy tắc đặt tên: bắt đầu bằng chữ hoặc \_, không có khoảng trắng, phân biệt hoa thường, không trùng từ khóa.

### b. Cấu trúc điều khiển trong Python

* Cấu trúc rẽ nhánh: if, elif, else
* Cấu trúc lặp: for, while
* Câu lệnh điều khiển vòng lặp: break, continue, pass

### c. Giải thích các cú pháp lệnh và phương thức sử dụng trong chương trình

| Lệnh / Phương thức | Ý nghĩa chính |
| --- | --- |
| input(), print() | Nhập và xuất dữ liệu |
| int(), float(), str() | Ép kiểu dữ liệu |
| for, while, if, else | Cấu trúc điều khiển |
| range(), math.sqrt() | Tạo dãy, tính căn bậc hai |
| split(), join(), append() | Xử lý chuỗi, danh sách |
| count(), dict(), re.search() | Đếm, tạo từ điển, tìm mẫu ký tự |

### 4.4. Tài liệu tham khảo

**Giáo trình Python cơ bản**, Viện Kỹ thuật và Công nghệ – Trường Đại học Vinh, 2025.

**Python Documentation** – The Python Software Foundation, <https://docs.python.org/3/>

**W3Schools – Python Tutorial**, <https://www.w3schools.com/python/>

**Bài 3 .Lập trình hàm trong Python**

**3.1. Mục đích**

Giúp sinh viên nắm bắt các kiến thức trong lập trình python sử dụng hàm.

* 1. **. sở lý thuyết**

Xem các quy tắc khai báo hàm, gọi hàm, giá trị trả về, tham số truyền vào, tham số mặc định, phạm vi của biến trong python, sử dụng các hàm có sẵn trong các thư viện của python

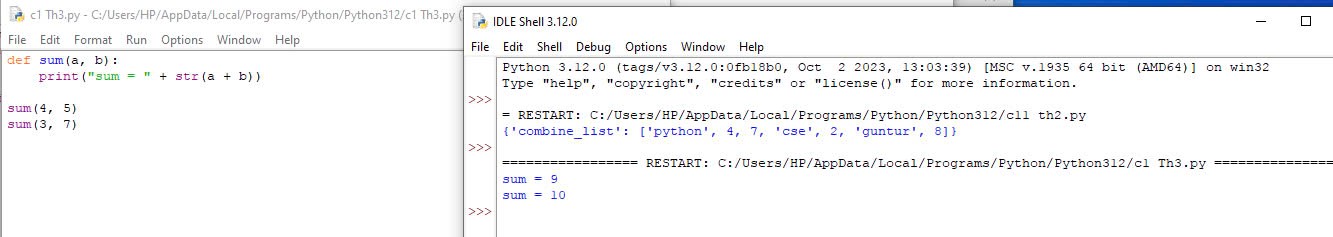
**3.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao**

Thiết bị thực hành thí nghiệm: Máy tính Vật tư tiêu hao: Không

**3.4. Các bước tiến hành**

**Thực hiện các bài tập sau sử dụng python**

1. **Viết hàm sum() tính tổng hai số**



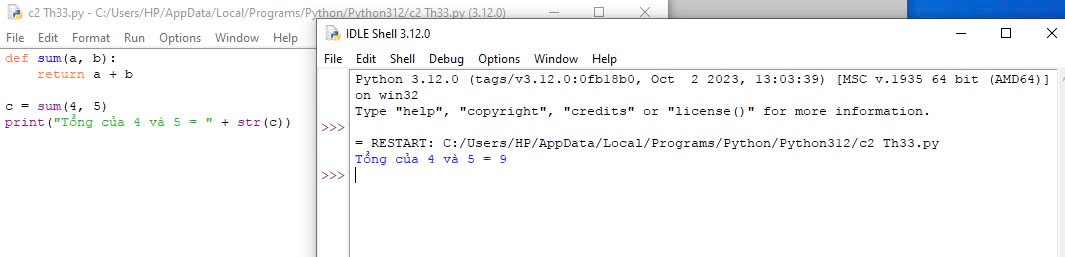
1. **. Viết hàm sum () với kết quả trả về**

def sum(a, b):

return a+ b

c = sum(4, 5);

print("Tong cua 4 va 5 = " + str(c))

****

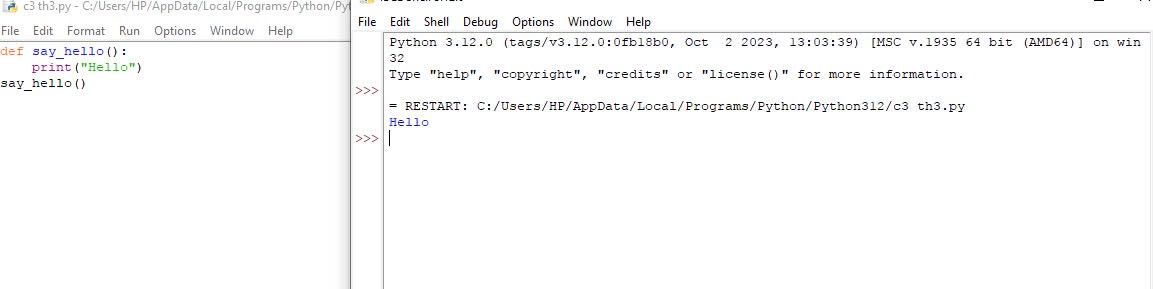
**3. Tìm và sửa lỗi chương trình**

def say\_hello():

a = "Hello"

print(a)

print(a)



**4.Viết chương trình có phạm vi biến như sau**

a = "Hello Guy!"

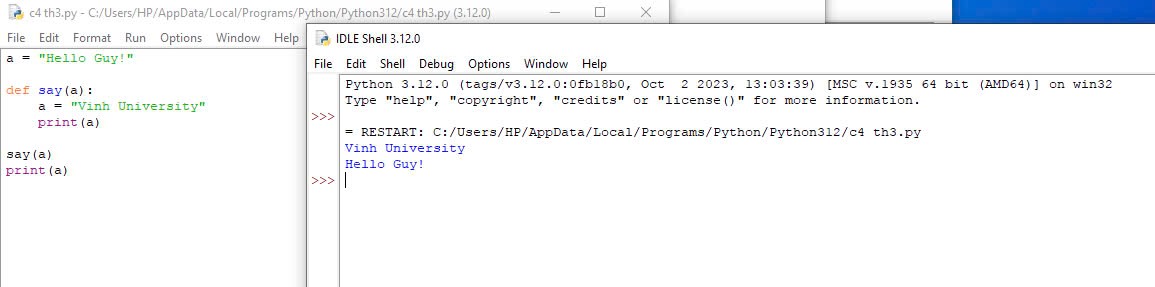
def say(a):

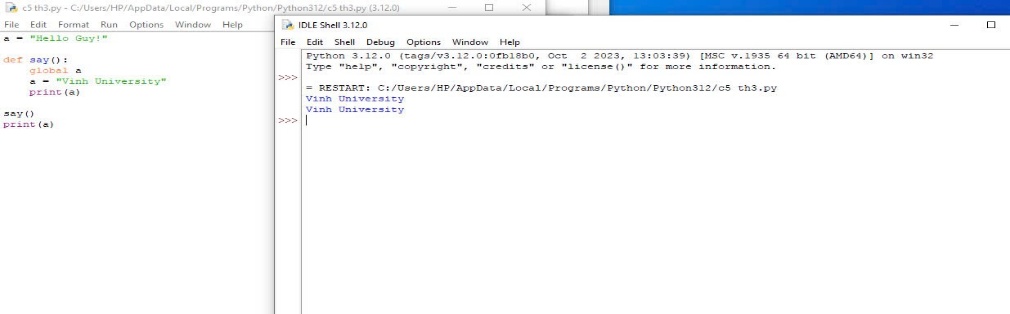
a = "Vinh University"

print(a)

say(a)

print(a)

****

**5****.Viết đoạn chương trình sau và theo dõi sự thay đổi của biến**

a = "Hello Guy!"

def say():

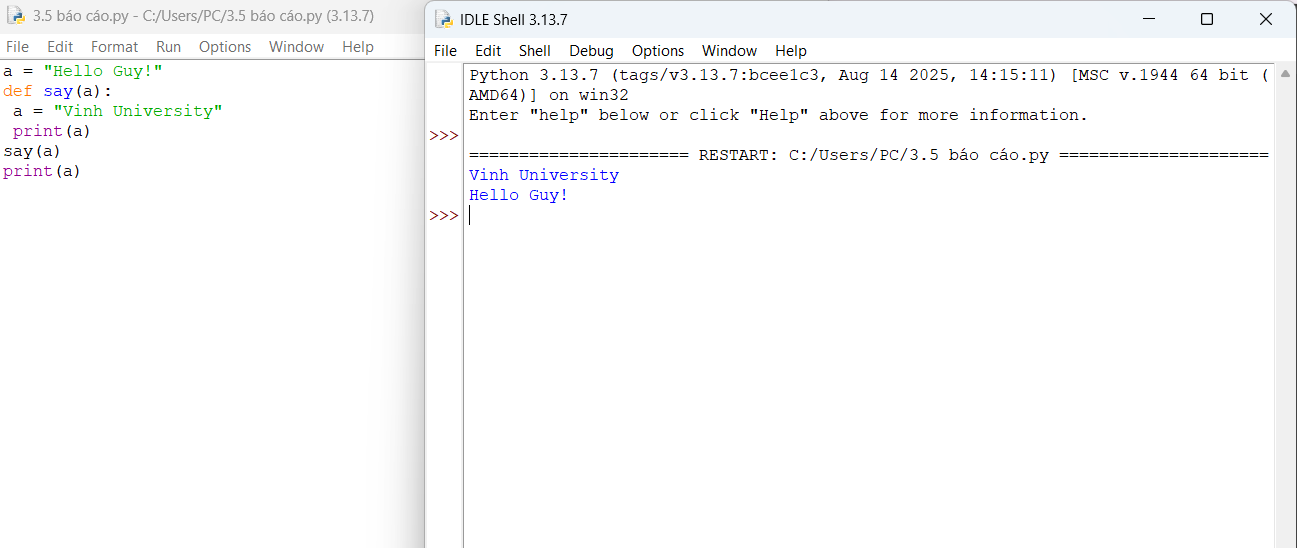
global a

a = " Vinh University "

print(a)

say()

(a)

****

**6.Viết chương trình sau và giải thích việc truyền tham số của hàm**

def get\_sum(\*num):

tmp = 0

# duyet cac tham so

for i in num:

tmp += i

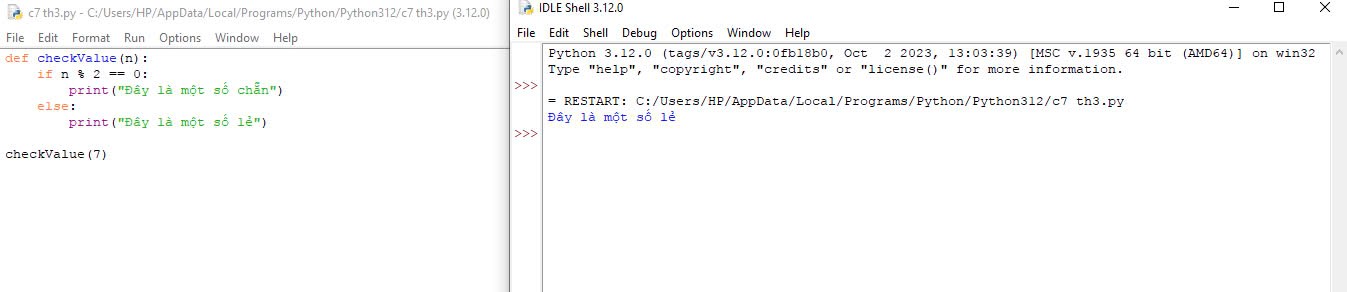
return tmp

result = get\_sum(1, 2, 3, 4, 5)

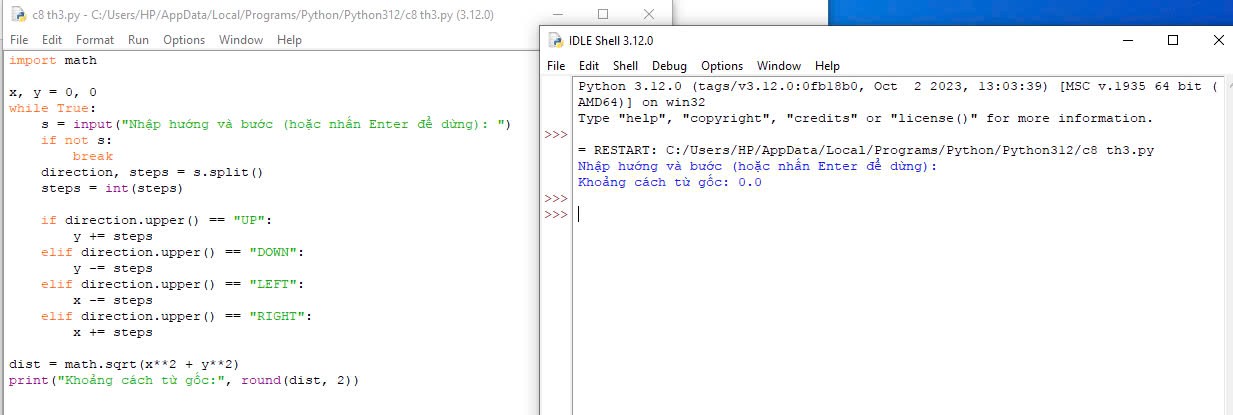
print(result)

****

**7.** **Định nghĩa hàm có thể chấp nhận input là số nguyên và in "Đây là một số chẵn" nếu nó chẵn và in "Đây là một số lẻ" nếu là số lẻ.**

****

**8. Một Robot di chuyển trong mặt phẳng bắt đầu từ điểm đầu tiên 0,0). Robot có thể.Di chuyển theo hướng UP, DOWN, LEFT và RIGHT với những bước nhất định.**

****

**9. Chương trình máy tính thực hiện các phép tính đơn giản**

“”” Program make a simple calculator that can add, subtract, multiply and

divide using functions”””

# This function adds two numbers

def add(x, y):

return x + y

# This function subtracts two numbers

def subtract(x, y):

return x - y

# This function multiplies two numbers

def multiply(x, y):

return x \* y

# This function divides two numbers

def divide(x, y):

return x / y

print("Select operation.")

print("1.Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

# Take input from the user

choice = input("Enter choice(1/2/3/4):")

num1 = int(input("Enter first number: "))

num2 = int(input("Enter second number: "))

if choice == '1':

print(num1,"+",num2,"=", add(num1,num2))

elif choice == '2':

print(num1,"-",num2,"=", subtract(num1,num2))

elif choice == '3':

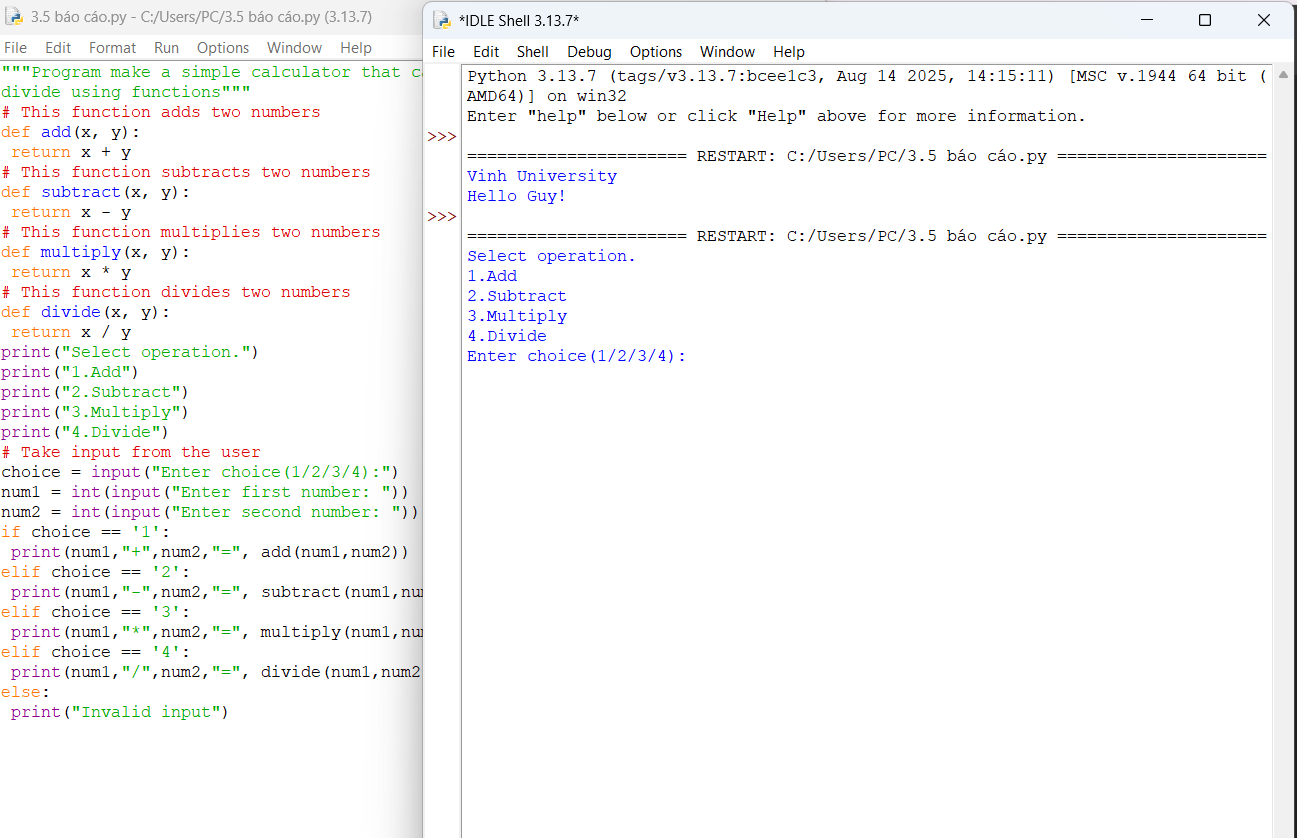
print(num1,"\*",num2,"=", multiply(num1,num2))

elif choice == '4':

print(num1,"/",num2,"=", divide(num1,num2))

else:

print("Invalid input")

****

**3.5 Câu hỏi kiểm tra**

Các quy tắc khai báo hàm, gọi hàm, giá trị trả về, tham số truyền vào, tham. Số mặc định, phạm vi của biến trong python, sử dụng các hàm có sẵn trong các thư.Viện của python trong các chương trình đã thực thi:

a. Khai báo hàm

Trong Python, hàm được tạo bằng từ khóa def, sau đó là tên hàm và các tham số.

Ví dụ:

def add(x, y):

  return x + y

Hàm này dùng để cộng hai số.

b. Gọi hàm

Sau khi tạo hàm, muốn dùng thì gọi tên nó, và truyền giá trị vào cho tham số.

Ví dụ:

add(5, 3)

c. Tham số và giá trị trả về

Tham số là những biến xuất hiện trong phần khai báo hàm (ví dụ x, y).

Khi gọi hàm, ta truyền vào các giá trị thật, gọi là đối số.

Từ khóa return giúp hàm trả lại kết quả để sử dụng ở nơi khác.

Nếu hàm không có return thì nó không trả gì cả.

d. Tham số mặc định

Python hỗ trợ cho tham số có sẵn giá trị.

Ví dụ:

def greet(name="student"):

  print("Hello", name)

Nếu gọi greet() thì nó sẽ dùng name = "student".

e. Phạm vi của biến

Biến tạo trong hàm chỉ dùng được trong hàm (biến cục bộ).

Biến tạo bên ngoài có thể dùng trong toàn chương trình (biến toàn cục).

f. Sử dụng hàm trong thư viện Python

Muốn dùng hàm có sẵn trong thư viện thì phải import.

Ví dụ dùng thư viện math:

import math

math.sqrt(25)

Hàm sqrt() dùng để tính căn bậc hai.

**Bài 4. Các kiểu dữ liệu có cấu trúc trong Python**

**4.1. Mục đích**

Giúp sinh viên nắm bắt các kiểu dữ liệu có cấu trúc trong lập trình python.

4.2. Cơ sở lý thuyết

Xem các quy tắc sử dụng các kiểu dữ liệu: chuỗi, số, list, tuple, set và dictionary

trong python.

**4.3. Thiết bị thực hành thí nghiệm và vật tư tiêu hao**

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

**4.4. Các bước tiến hành**

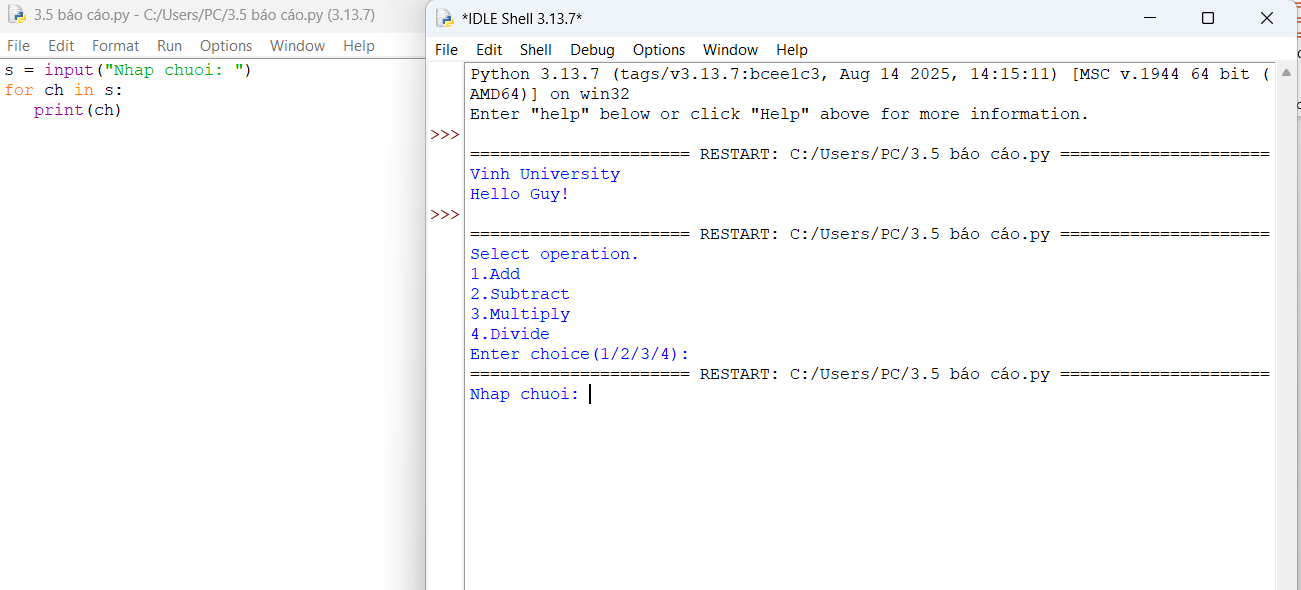
Thực hiện các bài tập sau sử dụng python

1.Nhập chuỗi S và in ra từng kí tự của S, mỗi kí tự trên một dòng.

s = input("Nhap chuoi: ")

for ch in s:

print(ch)

****

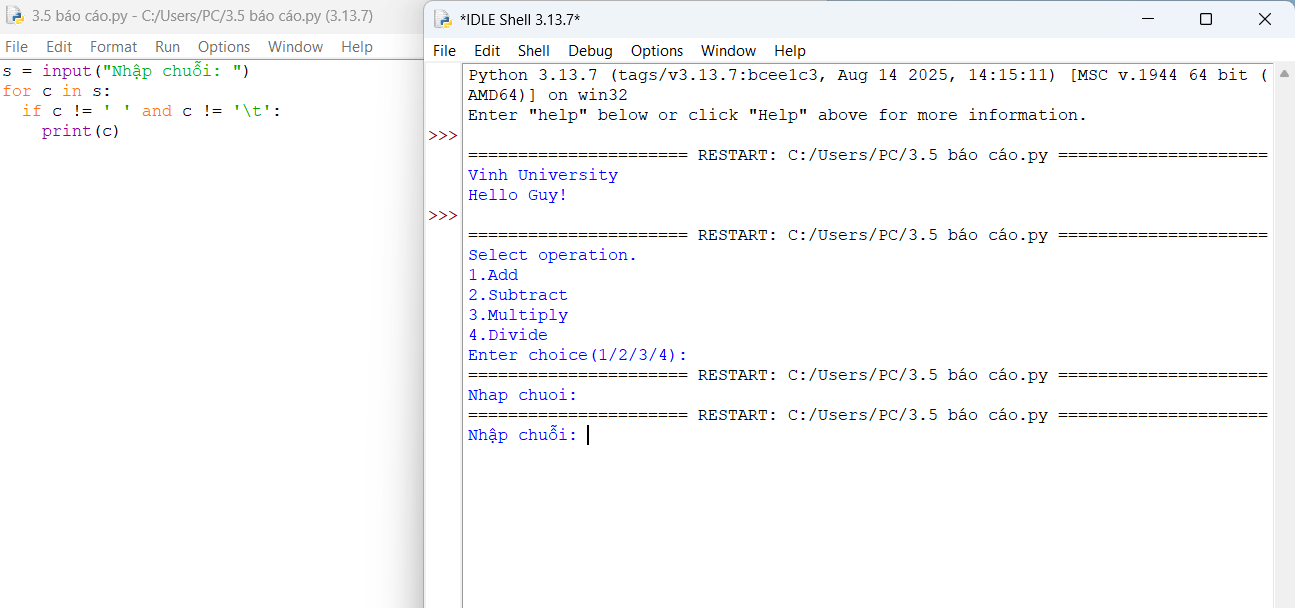
2. Chỉnh sửa ví dụ trên: hãy bỏ qua không in ra những kí tự “không nhìn thấy” (dấu space và dấu tab).

s = input("Nhập chuỗi: ")

for c in s:

if c != ' ' and c != '\t':

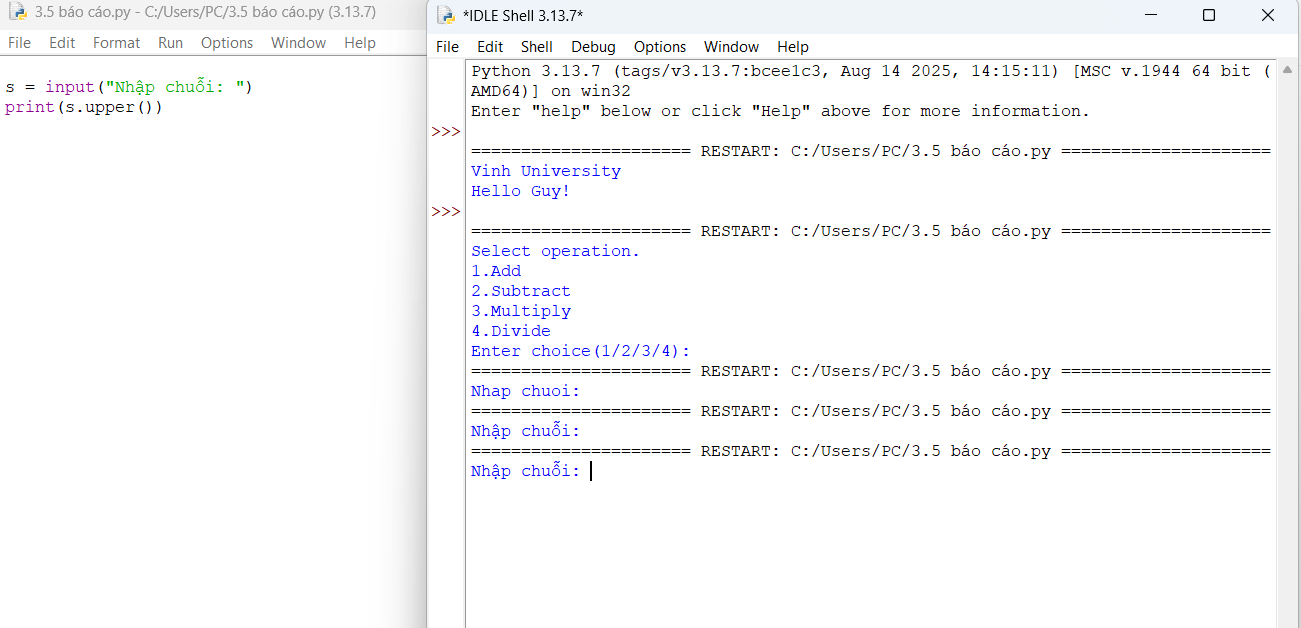
print(c)

****

3. Chỉnh sửa ví dụ ở bài 1: hãy các kí tự ở dạng IN HOA

s = input("Nhập chuỗi: ")

print(s.upper())

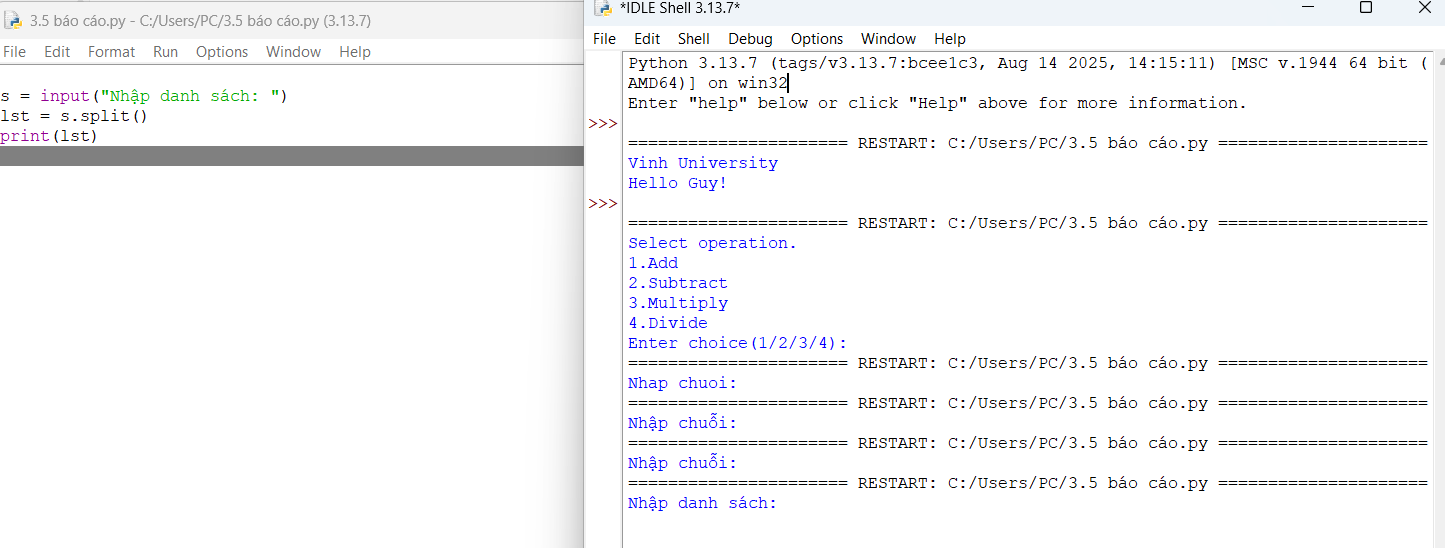
****

4. Nhập một danh sách trên một dòng, mỗi phần tử cách nhau bởi dấu trống hoặc tab, sau đó in ra dãy vừa nhập ra màn hình.

s = input("Nhập danh sách: ")

lst = s.split()

print(lst)

****

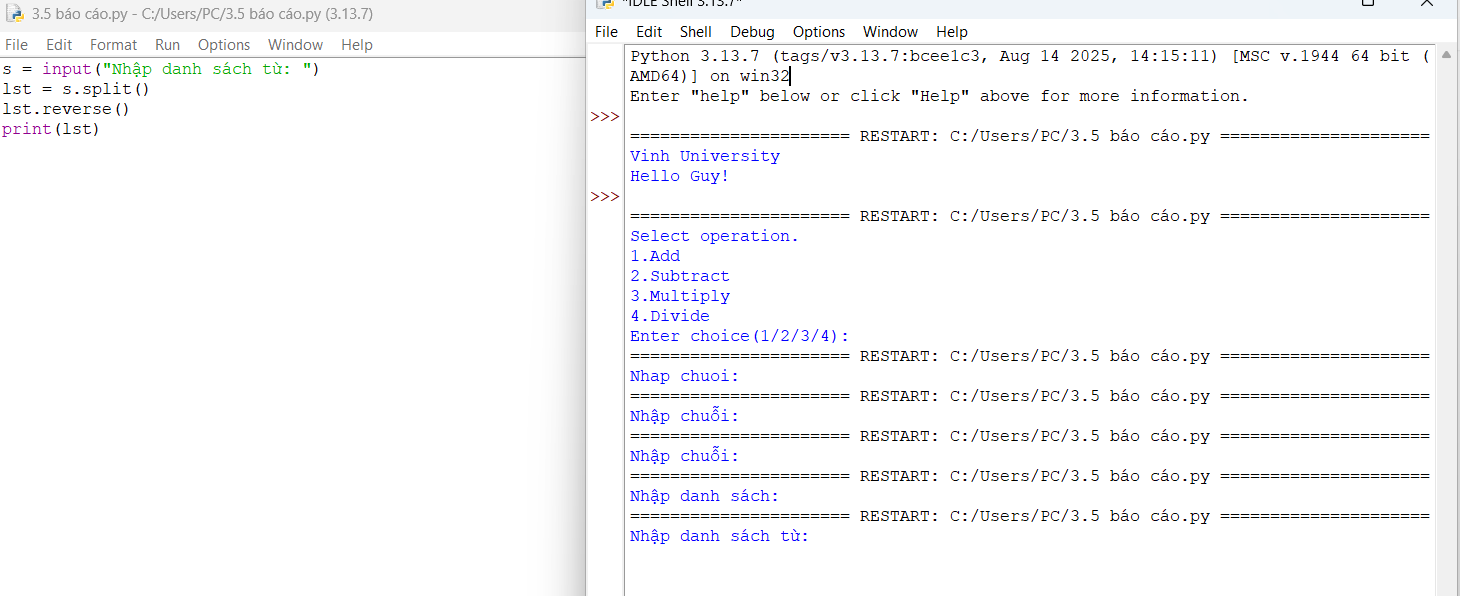
5. Chỉnh sửa ví dụ ở bài 4: nhập 1 danh sách các từ từ bàn phím, in ra các từ đó theo thứ tự ngược lại thứ tự vừa nhập (ví dụ nhập “mot hai ba” thì in ra “ba hai mot”)

s = input("Nhập danh sách từ: ")

lst = s.split()

lst.reverse()

print(lst)

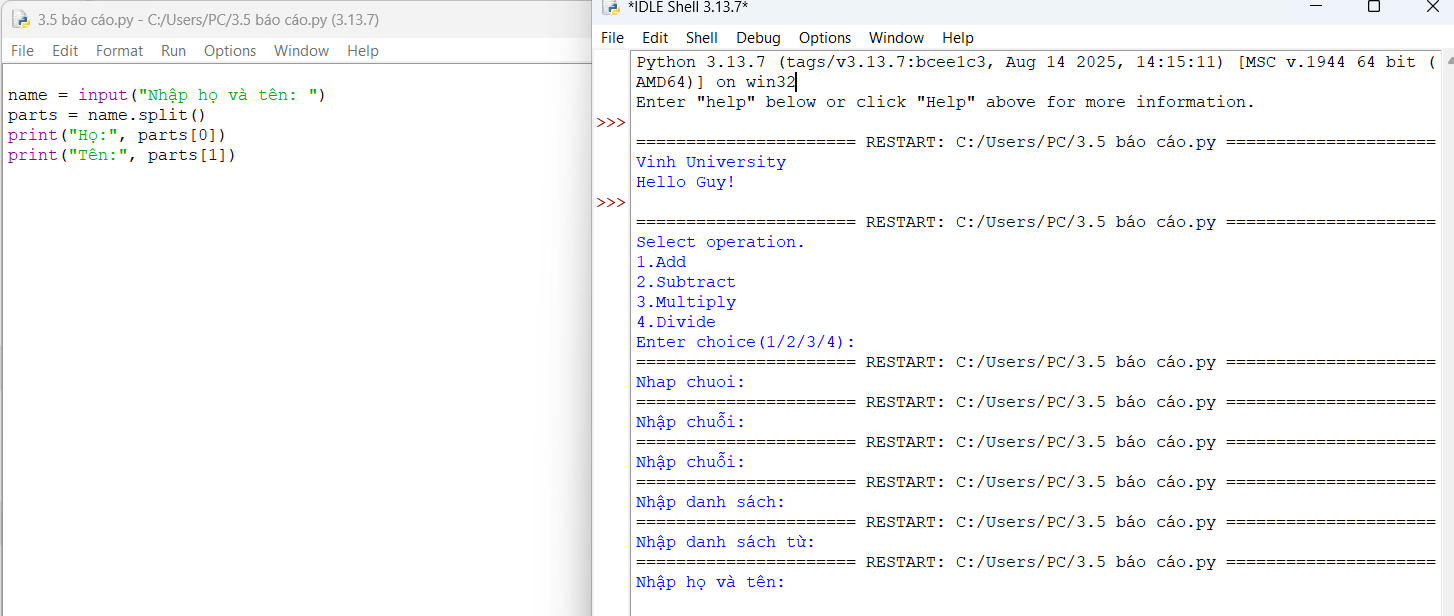
****

6. Nhập một tên người từ bàn phím, hãy tách phần họ và tên riêng của người đó và in chúng ra màn hình (giả thiết họ và tên riêng chỉ gồm một âm).   
  
 name = input("Nhập họ và tên: ")

parts = name.split()

print("Họ:", parts[0])

print("Tên:", parts[1])

  
  
  
 7. Nhập một chuỗi từ bàn phím, hãy loại bỏ tất cả các chữ số khỏi chuỗi và in lại nội dung chuỗi mới ra màn hình.  
  
s = input("Nhập chuỗi: ")  
kq = ""  
for c in s:  
  if not c.isdigit():  
  kq += c  
print(kq)

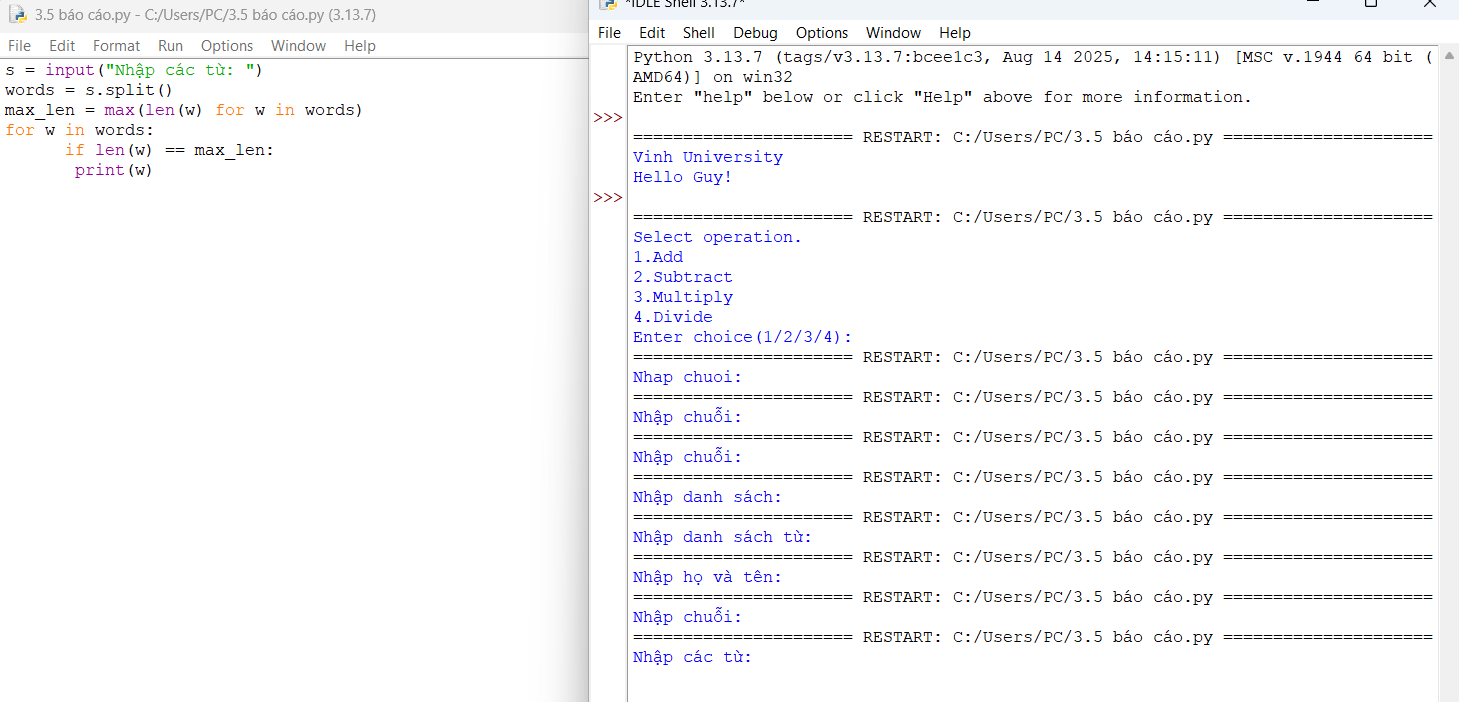
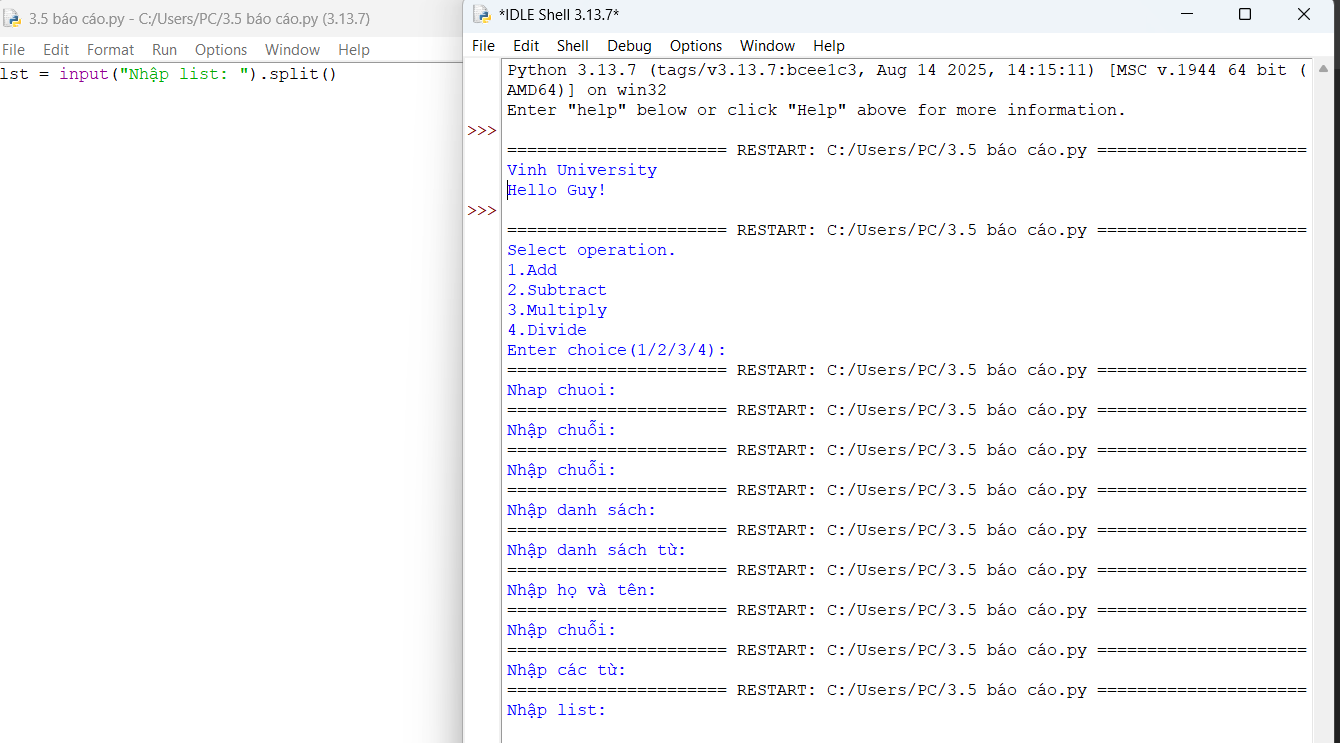
  
  
 8. Nhập một dãy các từ từ bàn phím, hãy in ra từ dài nhất trong dãy vừa nhập, in ra mọi từ có cùng độ dài nhất.   
  
s = input("Nhập các từ: ")

words = s.split()

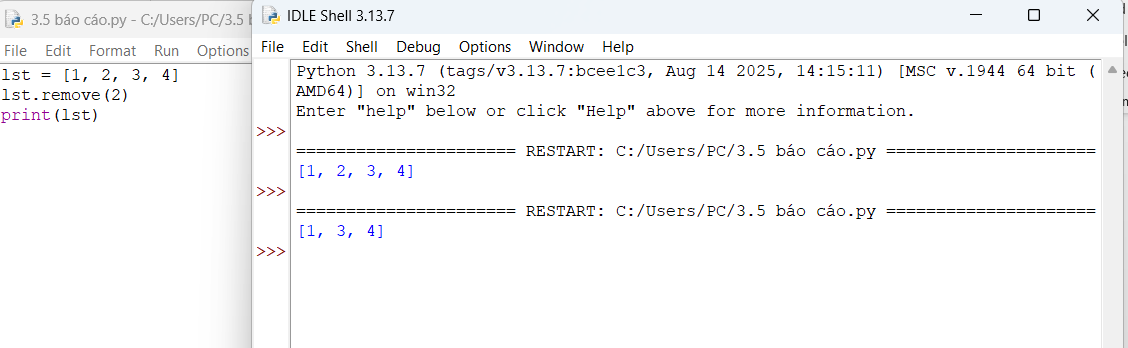
max\_len = max(len(w) for w in words)

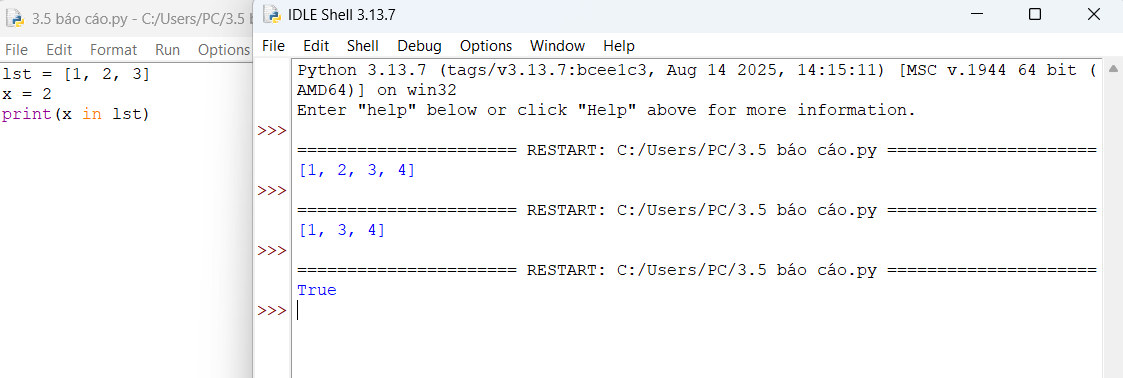
for w in words:

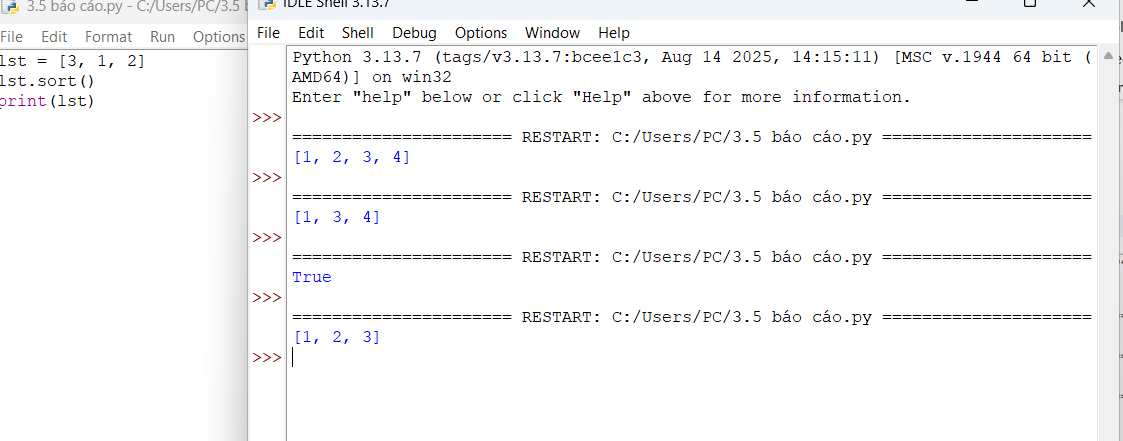
if len(w) == max\_len:

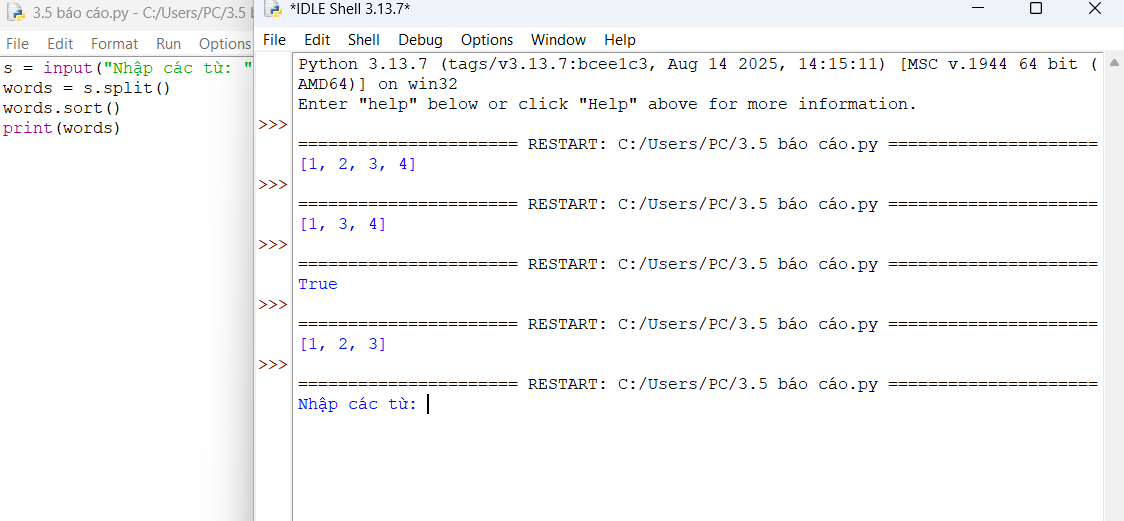
print(w)  
  
  
  
 9. Nhập một list từ bàn phím>  
   
lst = input("Nhập list: ").split()  
  
  
 10.Cắt list: lấy list nhưng bỏ phần tử đầu và cuối.  
  
lst = input("Nhập list: ").split()

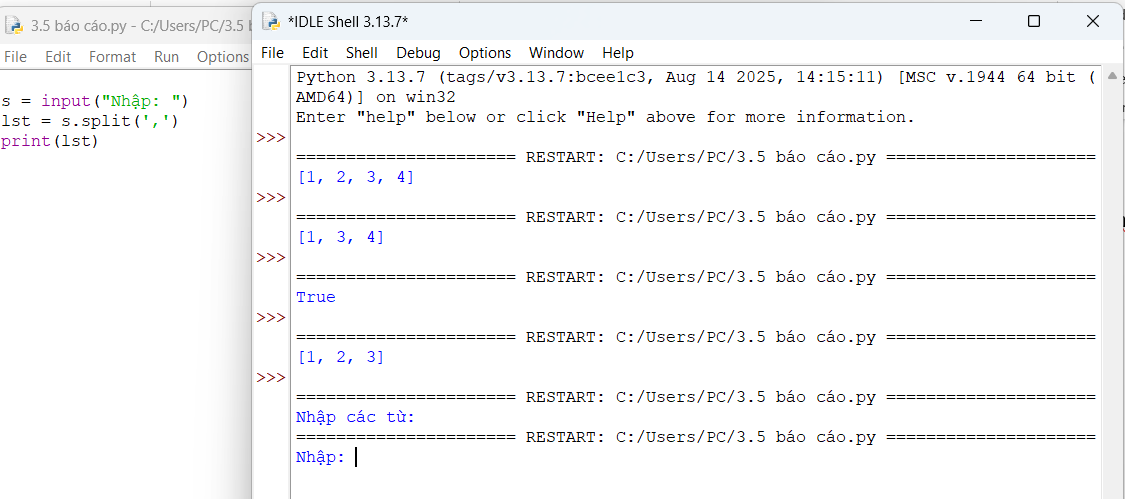
print(lst[1:-1])  
  
  
  
  
 11. Thêm phần tử vào list  
lst = [1, 2, 3]  
lst.append(4)  
print(lst)  
  


12. Bỏ phần tử khỏi list  
lst = [1, 2, 3, 4]  
lst.remove(2)  
print(lst)  
  


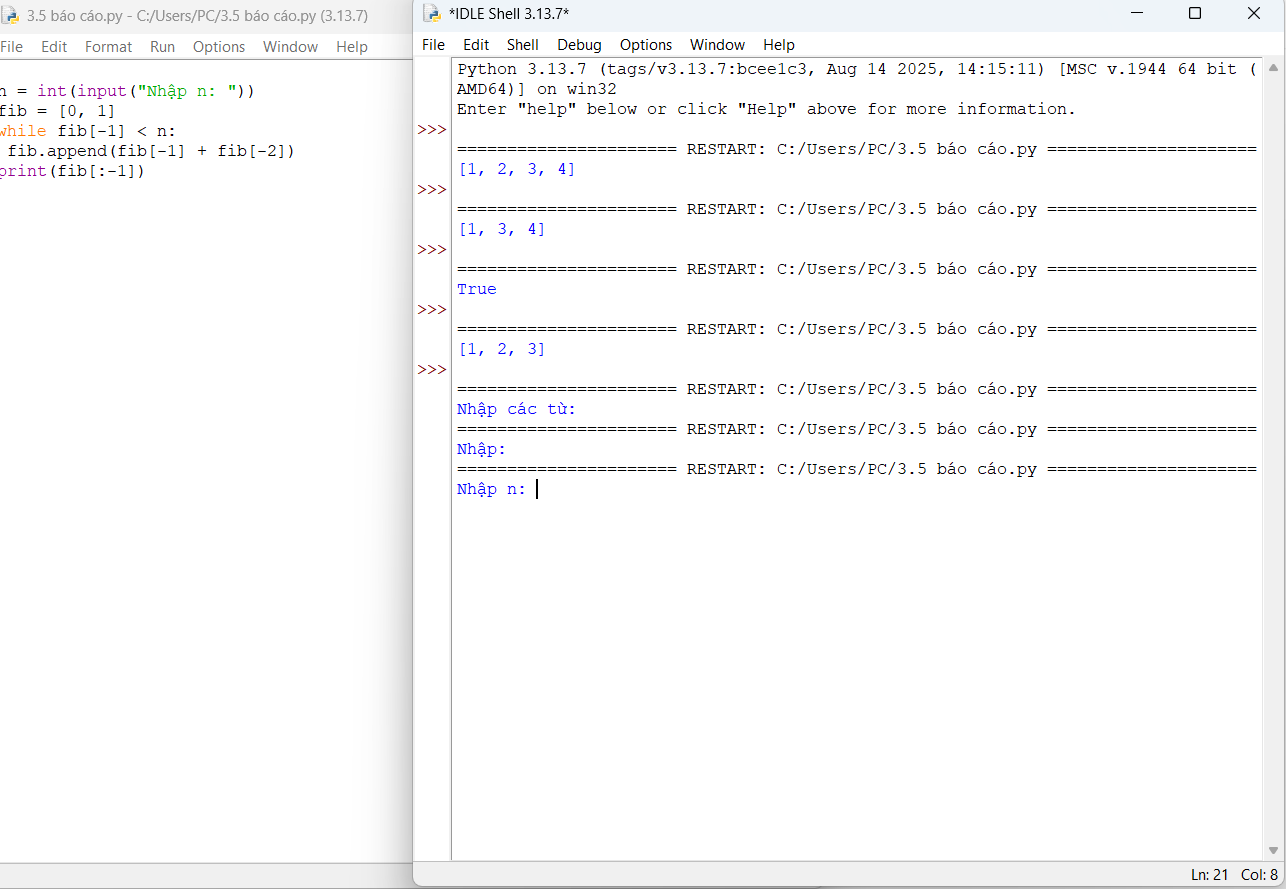
13. Tìm kiếm phần tử trong list  
lst = [1, 2, 3]  
x = 2  
print(x in lst)  
  


14. Sắp xếp các phần tử trong list  
lst = [3, 1, 2]  
lst.sort()  
print(lst)  
  
 

15. Người dùng nhập từ bàn phím liên tiếp các từ tiếng Anh viết tách nhau bởi dấu cách. Hãy nhập chuỗi đầu vào và tách thành các từ sau đó in ra màn hình các từ đó theo thứ tự từ điển.  
  
s = input("Nhập các từ: ")  
words = s.split()  
words.sort()  
print(words)  
  


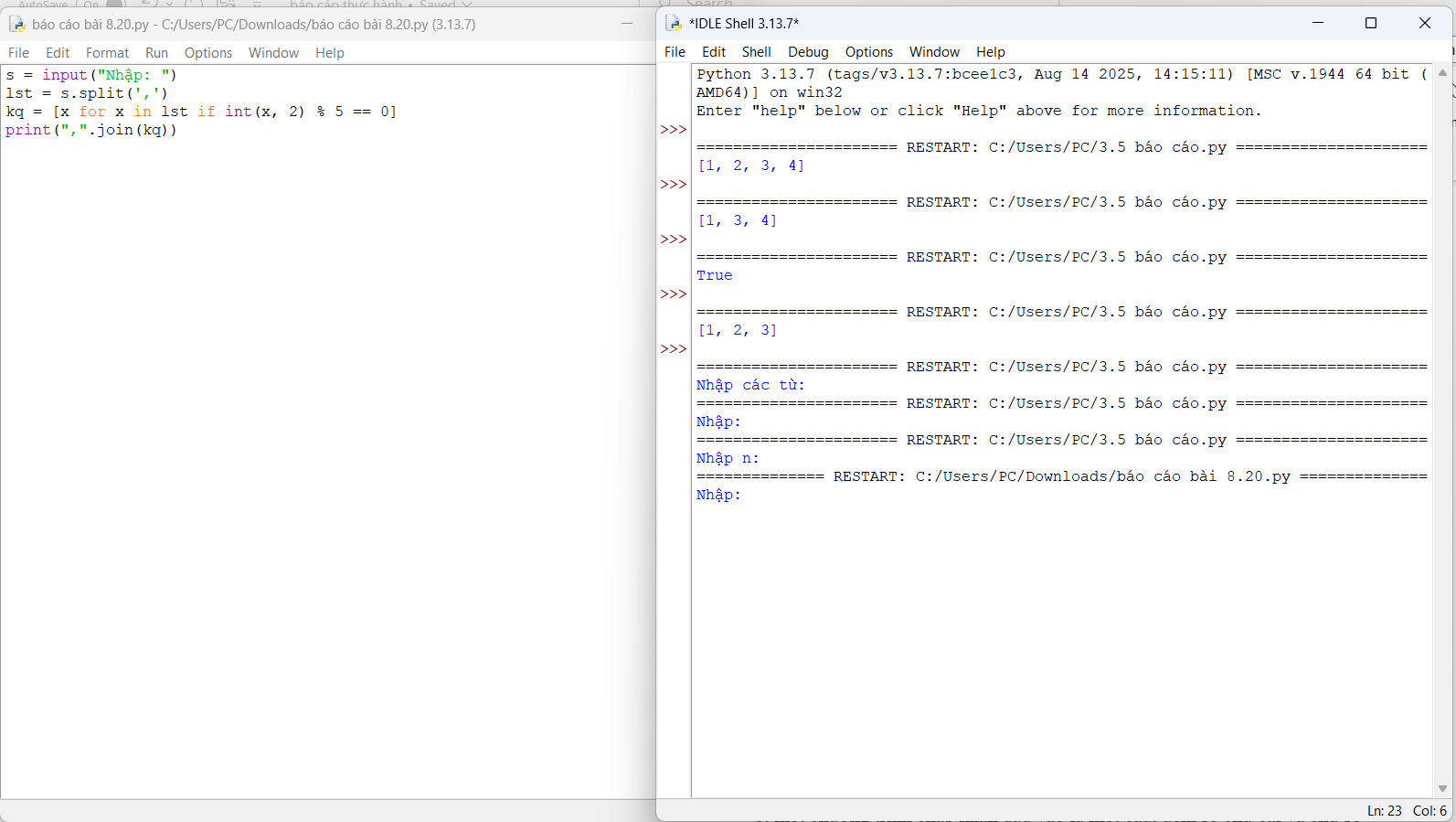
16. Người dùng nhập từ bàn phím chuỗi các số nhị phân viết liên tiếp được nối nhau bởi dấu phẩy. Hãy nhập chuỗi đầu vào sau đó in ra những giá trị được nhập.  
  
s = input("Nhập: ")  
lst = s.split(',')  
print(lst)  


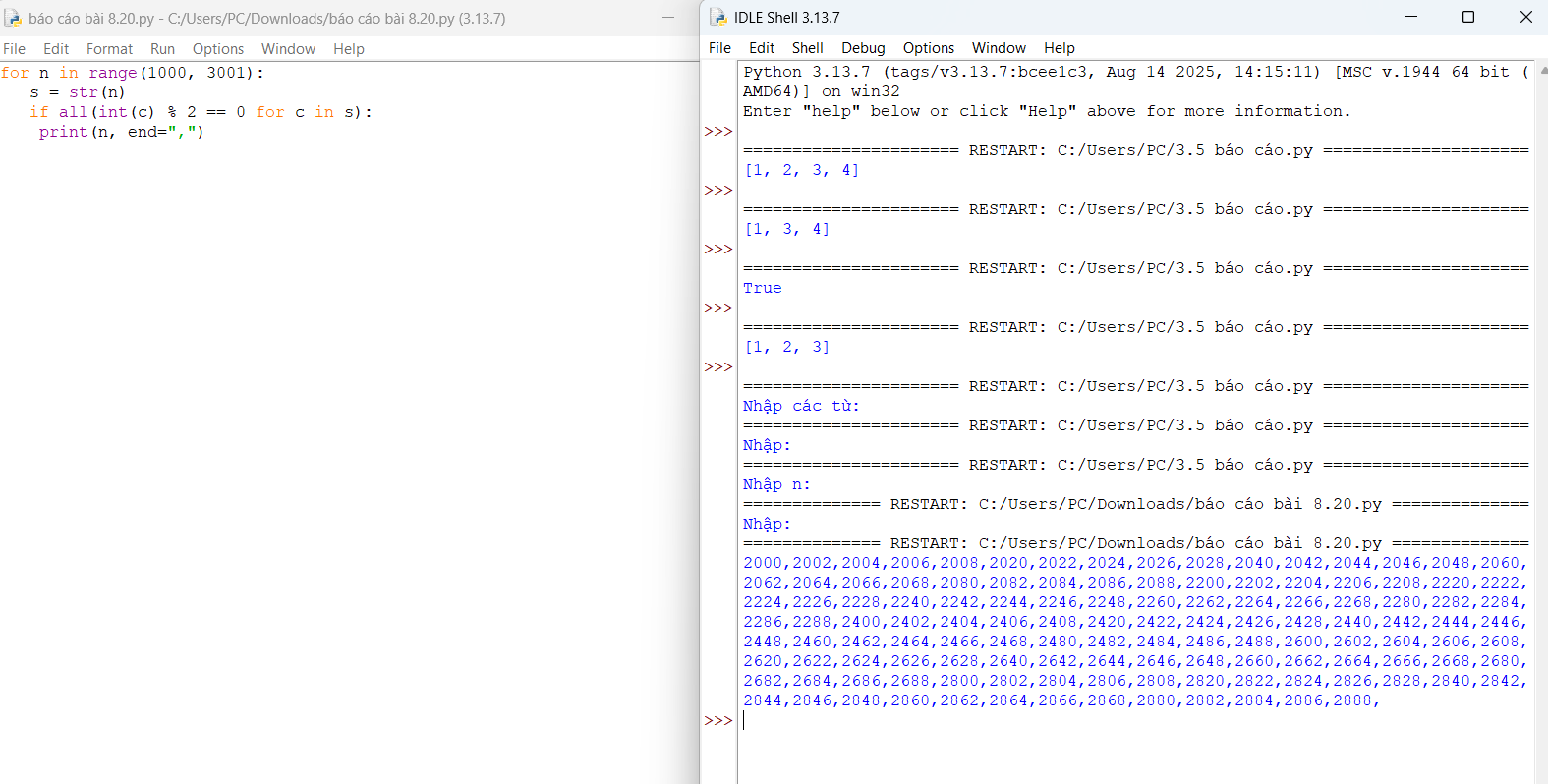
17. Nhập số n, in ra màn hình các số nguyên dương nhỏ hơn n có tổng các ước số lớn hơn chính nó.  
  
n = int(input("Nhập n: "))  
for x in range(1, n):  
 sum\_uoc = 0  
 for i in range(1, x):  
  if x % i == 0:  
   sum\_uoc += i  
 if sum\_uoc > x:  
  print(x)

18. Hãy nhập số nguyên n, tạo một list gồm các số fibonacci nhỏ hơn n và in ra màn hình.  
  
n = int(input("Nhập n: "))  
fib = [0, 1]  
while fib[-1] < n:  
 fib.append(fib[-1] + fib[-2])  
print(fib[:-1])  
  


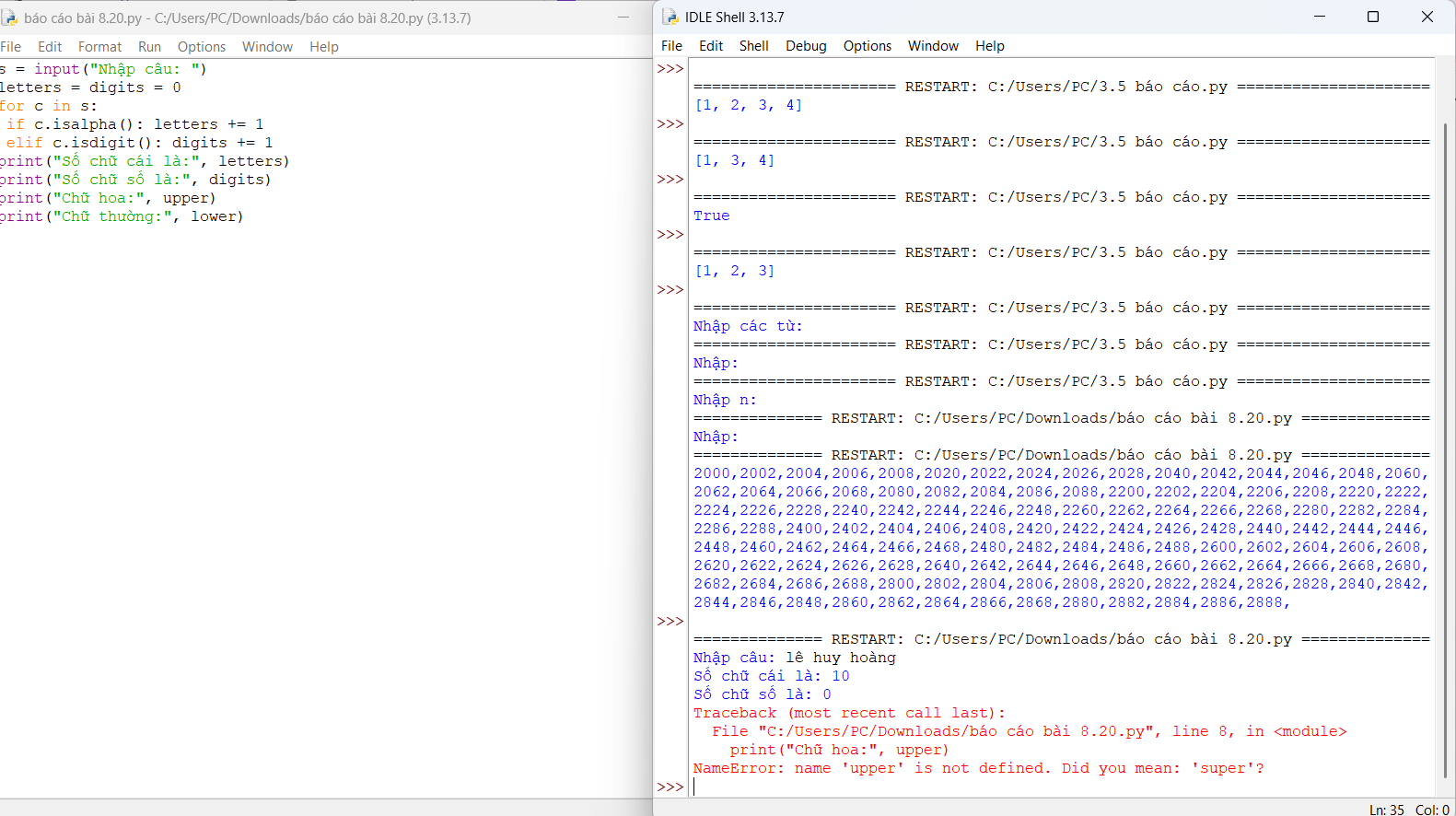
19. Hãy tạo ra tuple P gồm các số nguyên tố nhỏ hơn 1 triệu.  
  
import math  
lst = []  
for n in range(2, 1000000):  
 ok = True  
 for i in range(2, int(math.sqrt(n)) + 1):  
  if n % i == 0:  
   ok = False  
   break  
 if ok:  
  lst.append(n)  
P = tuple(lst)  
print(P)

20. Nhập n, in n dòng đầu tiên của tam giác pascal.  
  
n = int(input("Nhập n: "))  
tam\_giac = [[1]]  
for i in range(1, n):  
 hang = [1]  
 for j in range(1, i):  
  hang.append(tam\_giac[i-1][j-1] + tam\_giac[i-1][j])  
 hang.append(1)  
 tam\_giac.append(hang)  
for h in tam\_giac:  
 print(h)

21. Viết một chương trình chấp nhận đầu vào là chuỗi các số nhị phân 4 chữ số, phân tách bởi dấu phẩy, kiểm tra xem chúng có chia hết cho 5 không. Sau đó in các số chia hết cho 5 thành dãy phân tách bởi dấu phẩy.  
  
s = input("Nhập: ")  
lst = s.split(',')  
kq = [x for x in lst if int(x, 2) % 5 == 0]  
print(",".join(kq))  
  


22. Viết một chương trình tìm tất cả các số trong đoạn 1000 và 3000 sao cho tất cả các chữ số trong số đó là số chẵn. In các số tìm được thành chuỗi cách nhau bởi dấu phẩy, trên một dòng.  
  
for n in range(1000, 3001):  
 s = str(n)  
 if all(int(c) % 2 == 0 for c in s):  
  print(n, end=",")  
  


23. Viết một chương trình chấp nhận đầu vào là một câu, đếm số chữ cái và chữ số trong câu đó.  
  
s = input("Nhập câu: ")  
letters = digits = 0  
for c in s:  
 if c.isalpha(): letters += 1  
 elif c.isdigit(): digits += 1  
print("Số chữ cái là:", letters)  
print("Số chữ số là:", digits)

print("Chữ hoa:", upper)  
print("Chữ thường:", lower)  
  
  
  
**4.5. Tài liệu tham khảo**

Allen B. Downey, Think Python, O'Reilly Media, Inc, 2015  
  
**Bài 5. Thiết kế module trong Python**

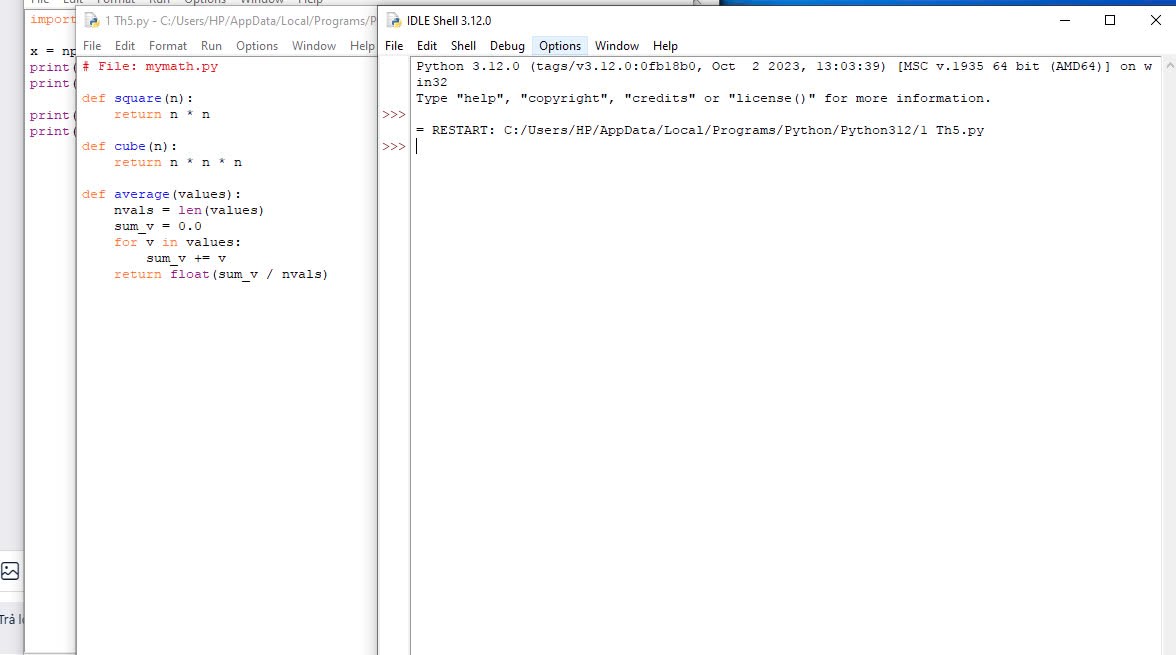
### 

### 5.1. Mục đích

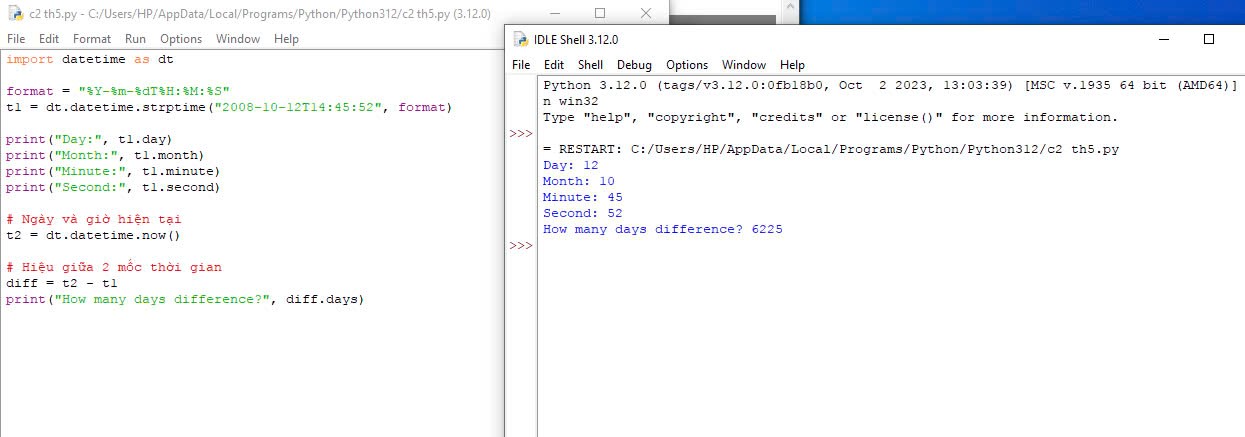
- Giúp học sinh nắm bắt việc thiết kế module trong lập trình Python, sử dụng module thư viện nampy trong các ứng dụng

**5.2. Các bước tiến hành**

1. Thực hiện module. Định nghĩa một module toán học gọi là mymath và sử dụng module này từ một tệp lệch riêng biệt.



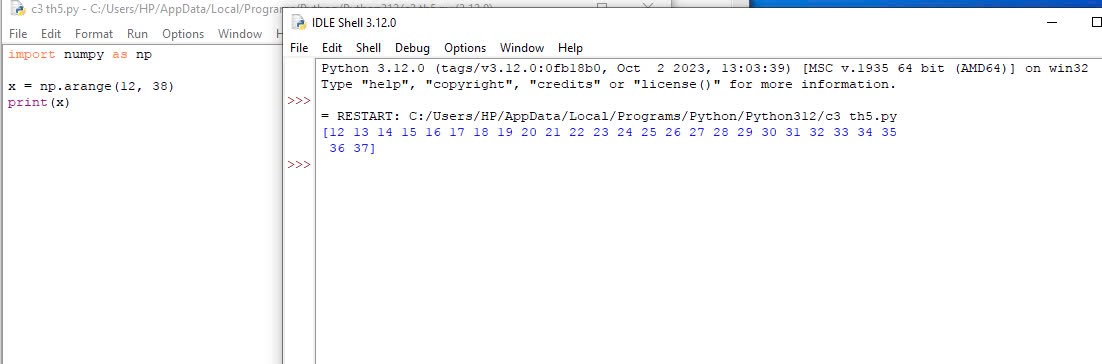
2. Sử dụng thư viện tiêu chuẩn của python ( date time )

****

3. Viết chương trình sử dụng thư viện NumPy để tạo một mảng với các giá trị nằm trong khoảng từ 12 đến 18

Đầu ra :

[ 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37]



4. Viếtchươngtrìnhđểtạomộtmảngvớicácgiátrịnằmtrongkhoảngtừ12đến38 và đảo ngược mảng đã tạo (phần tử đầu tiên trở thành cuối cùng)

Mảng được tạo:

[12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37] Mảng đảo ngược:

[37 36 35 34 33 32 31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12]

****

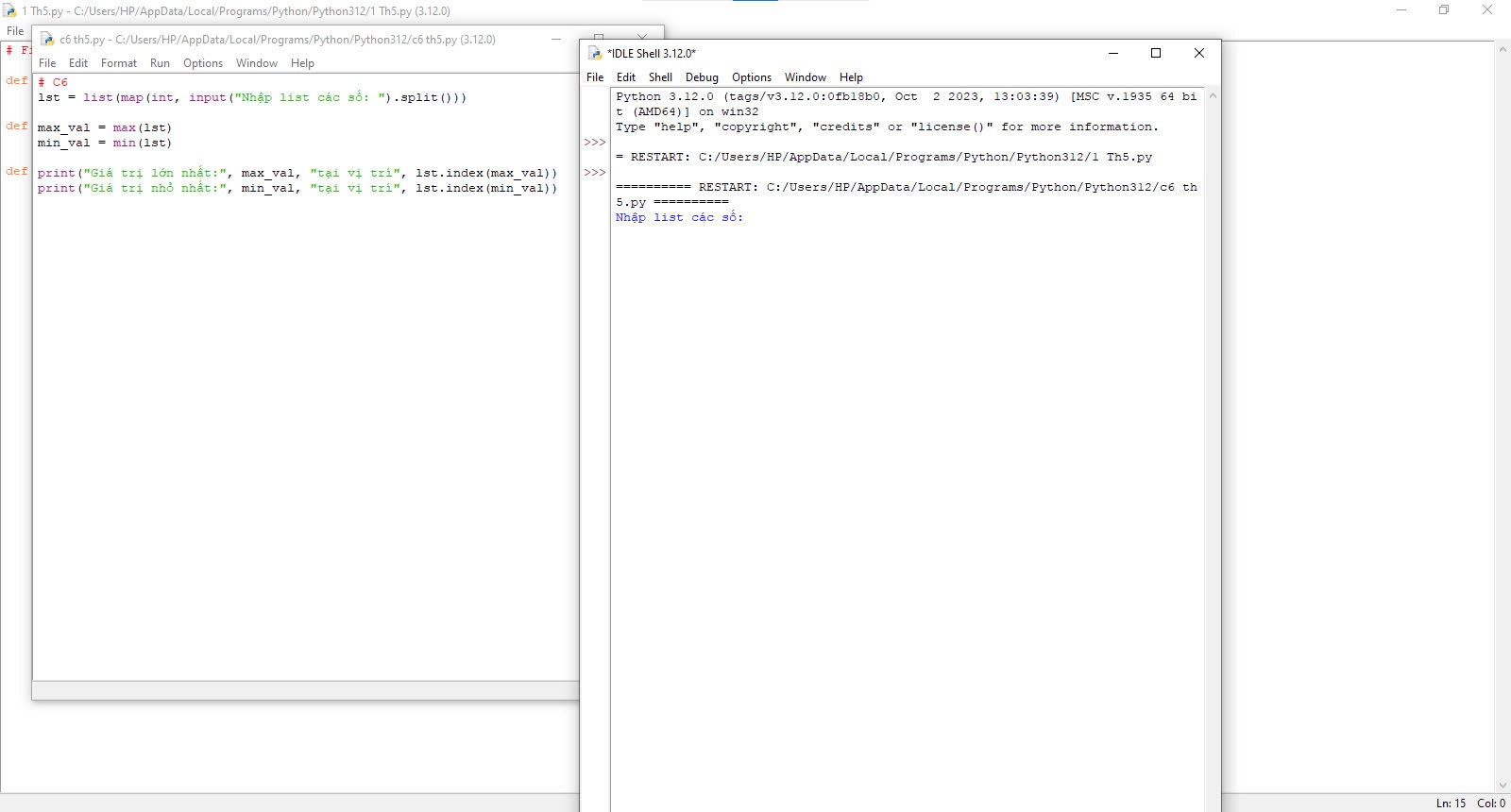
5. Viết chương trình tìm phần tử lớn nhất và nhỏ nhất của một danh sách

- Số lượng và giá trị của list được nhập từ bàn phím

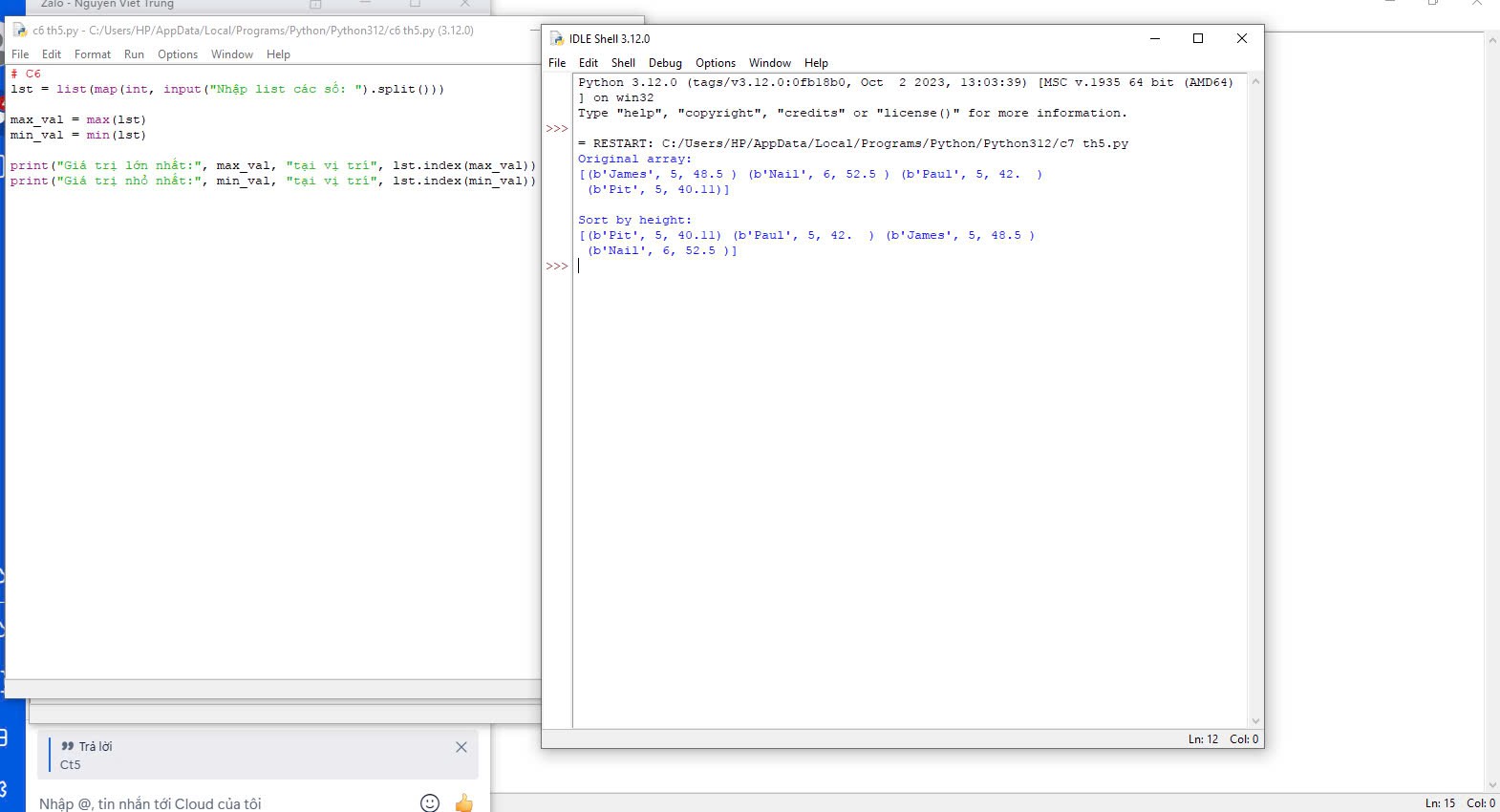
- Phương thức sắp xếp và tìm phần tử lớn nhất được viết thành module



6. In ra vị trí phần tử lớn nhất và nhỏ nhất tìm được ở bài tập trên

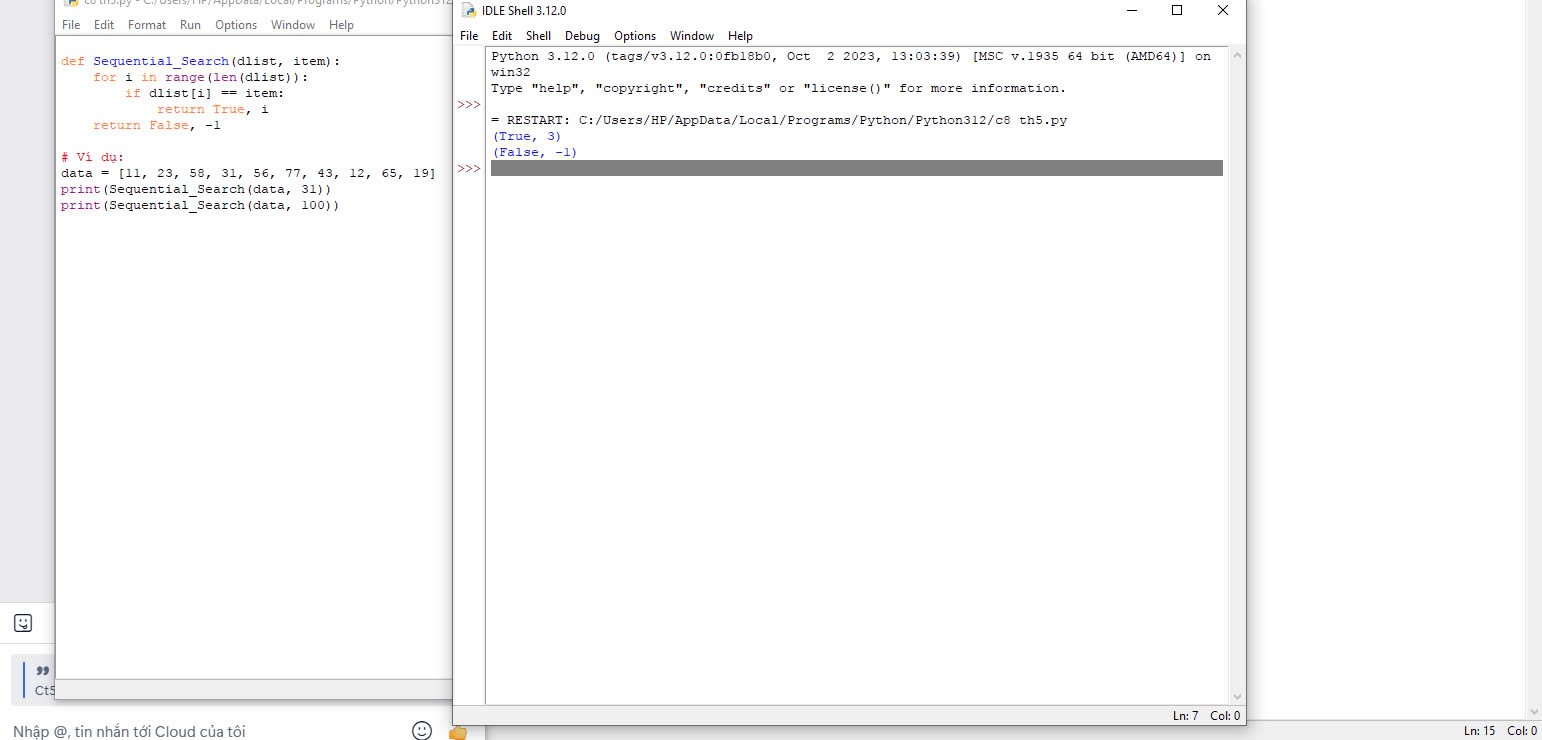


7. Viết chương trình sử dụng thư viện NumPy để tạo một mảng có cấu trúc từ tên sinh viên, chiều cao, lớp và các kiểu dữ liệu của họ. Bây giờ sắp xếp các mảng theo chiều cao.



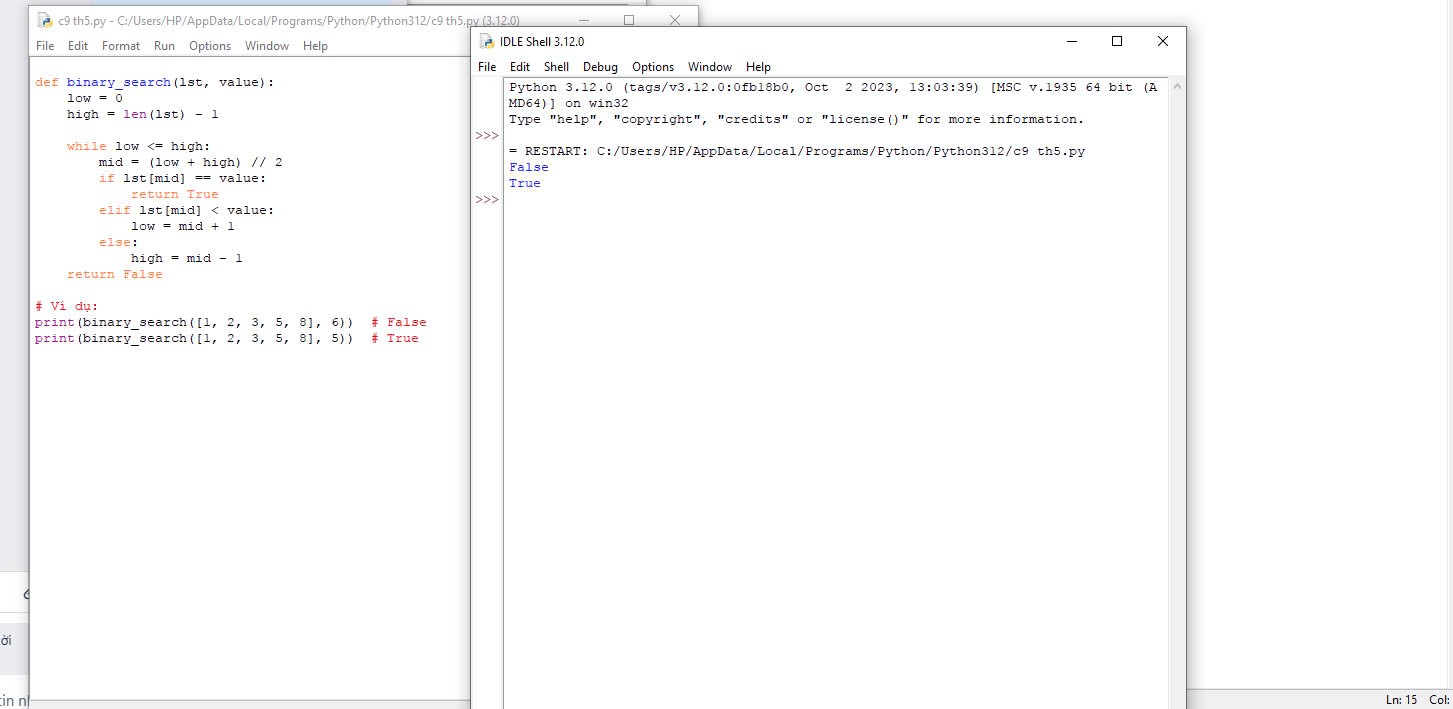
8. Xây dựng hàm “Sequential\_Search(dlist, item)” (giải thuật tìm kiếm tuyến tính) dưới dạng module. Viết chương trình nhập một dlist n phần tử từ bàn phím và tìm kiếm phần tử item bất kỳ.

Sequential\_Search([11,23,58,31,56,77,43,12,65,19],31) -> (True, 3)



9. Xây dựng hàm “binary\_search(list, value)” (giải thuật tìm kiếm nhị phân) dưới dạng module. Viết chương trình nhập một list n phần tử từ bàn phím và tìm kiếm phần tử value bất kỳ.

Binary\_search([1,2,3,5,8], 6) -> False binary\_search([1,2,3,5,8], 5) -> True



**5.3. Câu hỏi kiểm tra**

* Giải thích chương trình đã viết
* Tạo và sử dụng module trong Python

Module là một file .py chứa code (hàm, biến, lớp).

Dùng để chia chương trình lớn thành nhiều phần nhỏ, dễ quản lý và tái sử dụng.

Tạo module:

Ví dụ file mymath.py:

Def add(a, b):

Return a + b

Sử dụng module:

Import mymath

Print(mymath.add(3, 5))

* Các thuật toán sắp xếp, tìm kiếm

Một số thuật toán sắp xếp thông dụng

Bubble Sort – so sánh từng cặp phần tử, đổi chỗ, chậm nhưng dễ hiểu.

Selection Sort – tìm phần tử nhỏ nhất đưa lên đầu.

Sort – chèn phần tử vào vị trí đúng trong đoạn đã sắp

Quick Sort – chia để trị, rất nhanh trong thực tế.

Merge Sort – chia đôi dãy, trộn lại theo thứ tự.

Thuật toán tìm kiếm

Tìm kiếm tuần tự (Linear Search) – duyệt từng phần tử, đơn giản nhưng chậm.

Tìm kiếm nhị phân (Binary Search) – cực nhanh nhưng yêu cầu dữ liệu phải được sắp xếp.

* Thư viện NumPy

NumPy là thư viện mạnh để làm việc với mảng và tính toán khoa học.

Hỗ trợ:

Mảng n chiều (ndarray)

Toán tử hóa (tính rất nhanh)

Hàm thống kê, đại số tuyến tính

Sắp xếp, tìm kiếm, xử lý dữ liệu lớn

**5.4. Tài liệu tham khảo**

Allen B. Downey, Think Python, O’Reilly Media, Inc, 2015.

**Bài 6. Lập trình hướng đối tượng trong Python**

**6.1. Mục đích**

Giúp sinh viên năm bắt về lập trình hướng đối tượng trong Python.

**6.2. Các bước tiến hành**

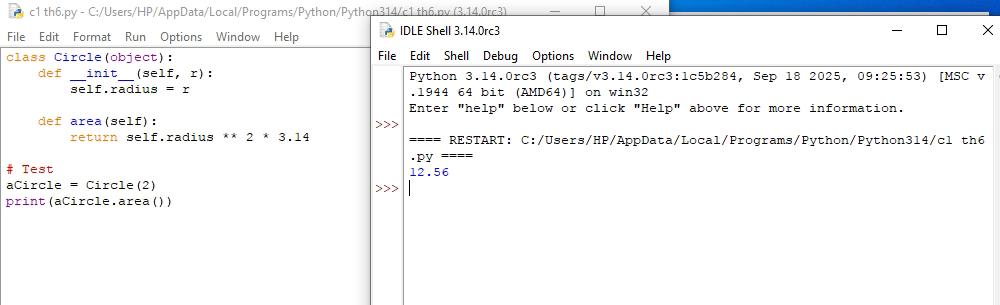
Thực hiện các bài tập sau sử dụng python

1. Định nghĩa một class có tên là Circle cót hể được xây dựng từ bán kính.Circlecó

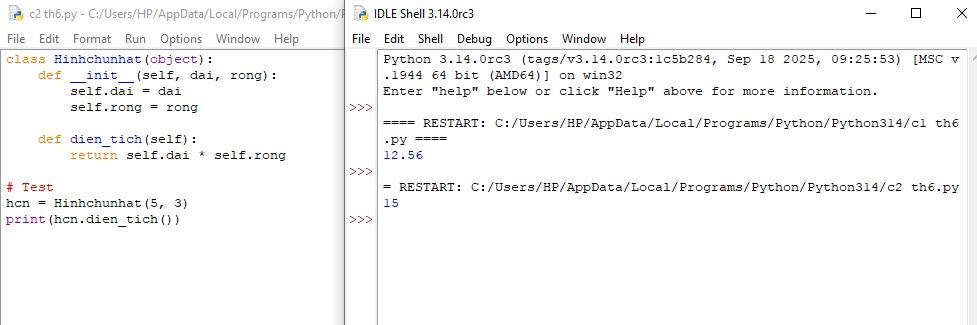
Một method có thể tính diện tích.

Gợi ý:

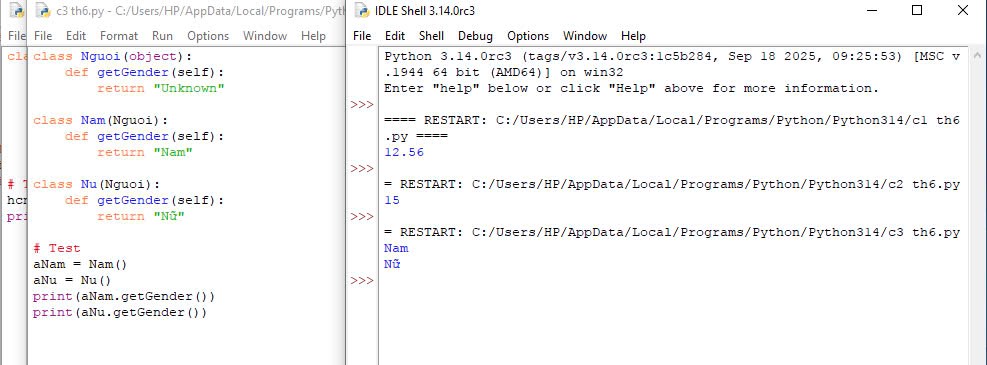
Sử dụng def methodName(self) để định nghĩa method.



1. Định nghĩa class có tên là Hinhchunhat được xây dựng bằng chiều dài và chiều rộng. Class Hinhchunhat có method để tính diện tích.

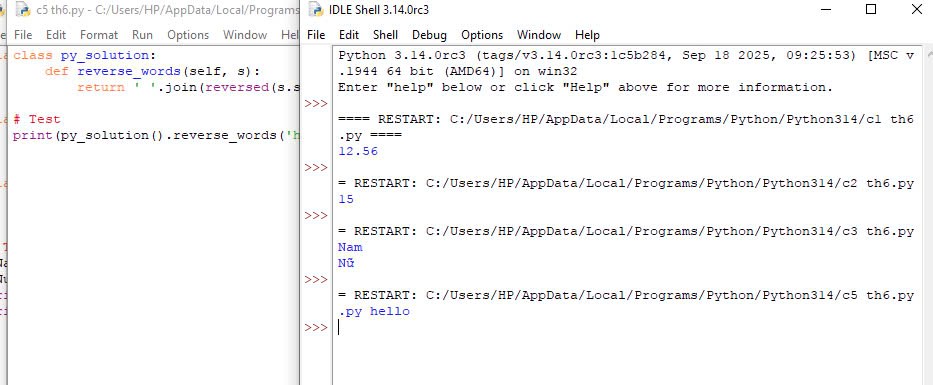


1. nghĩa class Nguoi và 2 class con của nó: Nam, Nu. Tất cả các class có method “getGender” có thể in “Nam” cho class Nam và “Nữ” cho class Nu.

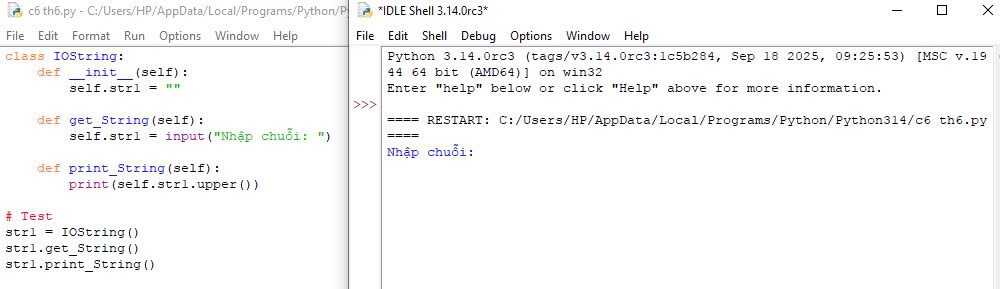


1. Viết chương trình Python dưới dạng class để đảo ngược chuỗi từ từng chữ.

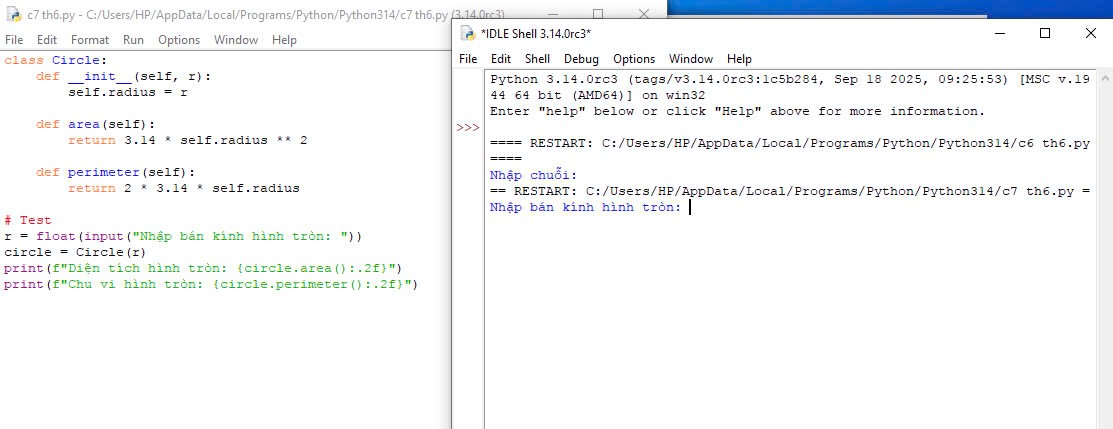
Dữ liệu vào : ‘hello .py’

Đầu ra : ‘.py hello’

1. Viết một class Python có hai phương thức get\_String và print\_String. Get\_String chấp nhận một chuỗi từ người dùng và print\_String in chuỗi đó bằng chữ in hoa.



1. Viết một class Python có tên Circle được xây dựng theo bán kính và hai phương thức sẽ tính diện tích và chu vi của hình tròn.



**6.3. Câu hỏi kiểm tra**

**-**  Giải thích chương trình đã viết

**-** Các khái niệm về lớp, đối tượng, thuộc tính, phương thức

• Lớp (Class)

Là một “bản thiết kế” dùng để mô tả một nhóm đối tượng có cùng đặc điểm.

• Đối tượng (Object)

Là thực thể được tạo ra từ lớp.

→ t1 là một đối tượng của lớp Bank.

• Thuộc tính (Attribute)

Là dữ liệu mô tả đặc điểm của đối tượng

• Phương thức (Method)

Là các hàm nằm trong lớp, dùng để mô tả hành vi của đối tượng

**6.4. Tài liệu tham khảo**

Allen B. Downey, Think Python, O’Reilly Media, Inc., 2015

**Bài 7: Thao tác trên tập tin và thư mục trong Python**

**7.1.Mục đích**

Giúp sinh viên có thể thao tác với các file văn bản sử dụng python.

**7.2.Cơ sở lý thuyết**

Xem các quy tắc mở, đóng file văn bản, đọc và ghi nội dung của các file.

* 1. **.Thiết bị thực hành thí nghiệm và vật tư tiêu hao**

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

**7.4. Các bước tiến hành**

Thực hiện các bài tập sau sử dụng python

1.Chương trình đọc file và in đảo ngược kết quả

input\_file = open('a.txt', 'r')

for line in input\_file:

line = line.rstrip('\n') # bỏ ký tự xuống dòng

reversed\_line = line[::-1] # đảo chuỗi

print(reversed\_line)

input\_file.close()

2. Chương trình đọc một file, tính số ký tự, số từ và số dòng của file

Char = 0

Words = 0

Lines = 0

With open(‘a.txt’, ‘r’) as f:

For line in f:

Lines += 1

Char += len(line)

Words += len(line.split())

Print(“Number of characters:”, char)

Print(“Number of words:”, words)

Print(“Number of lines:”, lines)

3. Chương trình Python để đọc n dòng đầu tiên của tệp

With open(‘a.txt’, ‘r’) as f:

Content = f.read()

Print(content)

4. Chương trình Python để đọc n dòng đầu tiên của tệp

From itertools import islice

Def read\_from\_head(fname, n):

With open(fname, ‘r’) as f:

For line in islice(f, n):

Print(line, end=’’)

Read\_from\_head(‘test.txt’, 2)

Chương trình Python để nối văn bản vào tệp và hiển thị văn bản.

Def file\_read(fname):

With open(fname, “a”) as f:

f.write(“\nPython Exercises”)

f.write(“\nJava Exercises”)

with open(fname, “r”) as f:

print(f.read())

file\_read(‘abc.txt’)

5. Chương trình Python để đọc n dòng cuối cùng của tệp

Def read\_from\_tail(fname, n):

With open(fname, ‘r’) as f:

Lines = f.readlines()

For line in lines[-n:]:

Print(line, end=’’)

Read\_from\_tail(‘test.txt’, 2)

6. Viết chương trình Python để đếm số dòng trong tệp văn bản

Count = 0

With open(‘a.txt’, ‘r’) as f:

For \_ in f:

Count += 1

Print(“Number of lines:”, count)

7. Viết chương trình Python để viết nội dung danh sách vào tệp.

My\_list = [“Python”, “Java”, “C++”, “JavaScript”]

With open(‘output.txt’, ‘w’) as f:

For item in my\_list:

f.write(item + ‘\n’)

print(“File written successfully!”)

8. Viết chương trình Python để sao chép nội dung của tệp này sang tệp khác.

Def copy\_file(src, dest):

With open(src, “r”, encoding=”utf-8”) as f1:

With open(dest, “w”, encoding=”utf-8”) as f2:

For line in f1:

F2.write(line)

Viết chương trình python để tìm những từ dài nhất trong văn bản

Def longest\_words(filename):

With open(filename, “r”, encoding=”utf-8”) as f:

Text = f.read()

# Tách từ, bỏ ký tự đặc biệt

Words = text.split()

Max\_len = max(len(word) for word in words)

Longest = [word for word in words if len(word) == max\_len]

Return set(longest) # Loại bỏ từ trùng lặp

**7.5. Câu hỏi kiểm tra**

Mở tệp: open()

Chế độ mở: “r”, “w”, “a”, “r+”

Đọc tệp: read(), readline(), readlines()

Ghi tệp: write(), writelines()

Đóng tệp: close()

Cách dùng with open(...) để tự động đóng tệp.

**7.6. Tài liệu tham khảo**

Allen B. Downey, Think Python, O’Reilly Media, Inc, 2015.

**Bài 8: Lập trình giao diện trong Python**

* 1. **Mục đích**

Giúp sinh viên làm quen với lập trìnhđồ họa và sử dụng thư viện turtle, Tkinter trong python.

* 1. **Cơ sở lý thuyết**

Xem các quy tắc xây dựng và quản lý layout, widget, hộp thoại, đồ họa trong python.

* 1. **Thiết bị thực hành thí nghiệm và vật tư tiêu hao**

Thiết bị thực hành thí nghiệm: Máy tính

Vật tư tiêu hao: Không

* 1. **Các bước tiến hành**

Thực hiện các bài tập thực hành sử dụng python:

1. Viết chương trình đồ họa sử dụng thư viện turtle, kiểm tra kết quả và giải thích chương trình

Import turtle

# Tạo cửa sổ

Window = turtle.Screen()

Window.bgcolor(“lightgreen”)

# Tạo bút vẽ

Painter = turtle.Turtle()

Painter.fillcolor(‘blue’)

Painter.pencolor(‘blue’)

Painter.pensize(3)

# Hàm vẽ hình vuông

Def drawsq(t, s):

For i in range(4):

t.forward(s)

t.left(90)

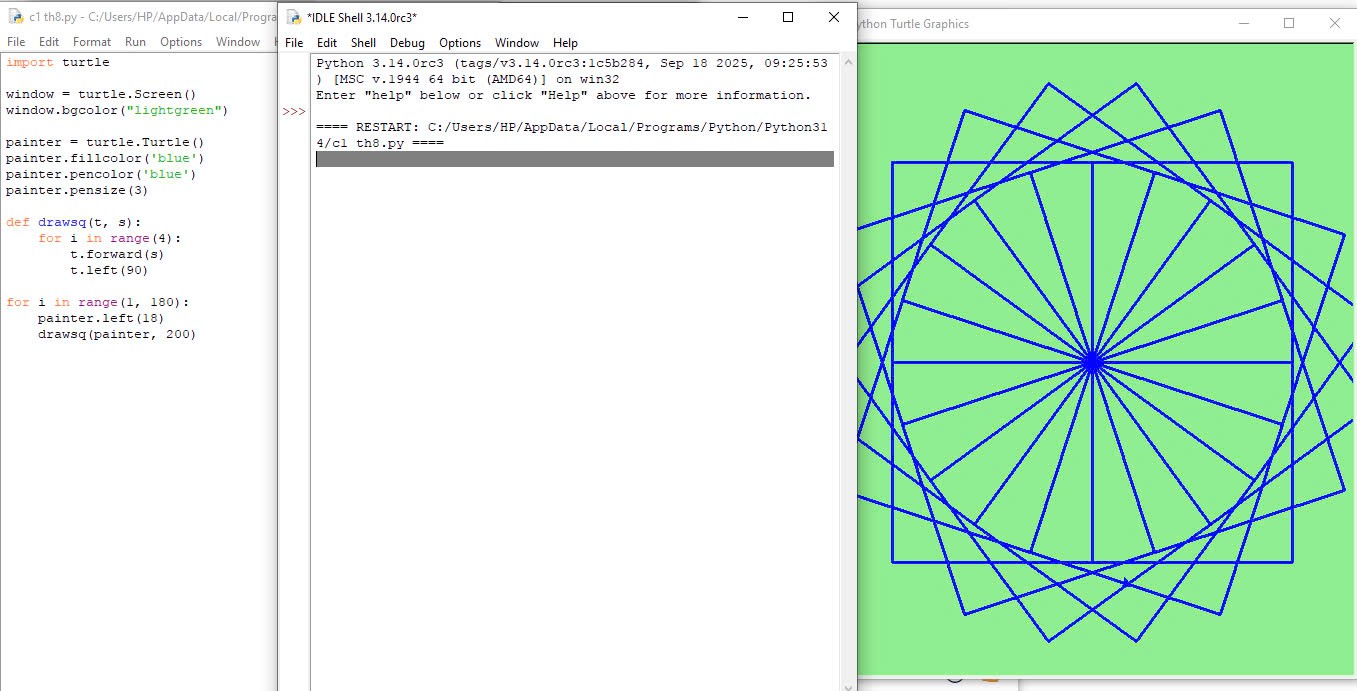
# Vẽ 10 hình vuông xoay dần

For i in range(10):

Painter.left(18)

Drawsq(painter, 200)

Turtle.done()



2.Viết chương trình đồ họa sử dụng thư viện turtle.

Import turtle

Import random

Colors = [“red”, “green”, “blue”, “orange”, “purple”, “pink”, “yellow”]

Painter = turtle.Turtle()

Painter.pensize(3)

For i in range(10):

Color = random.choice(colors)

Painter.pencolor(color)

Painter.circle(100)

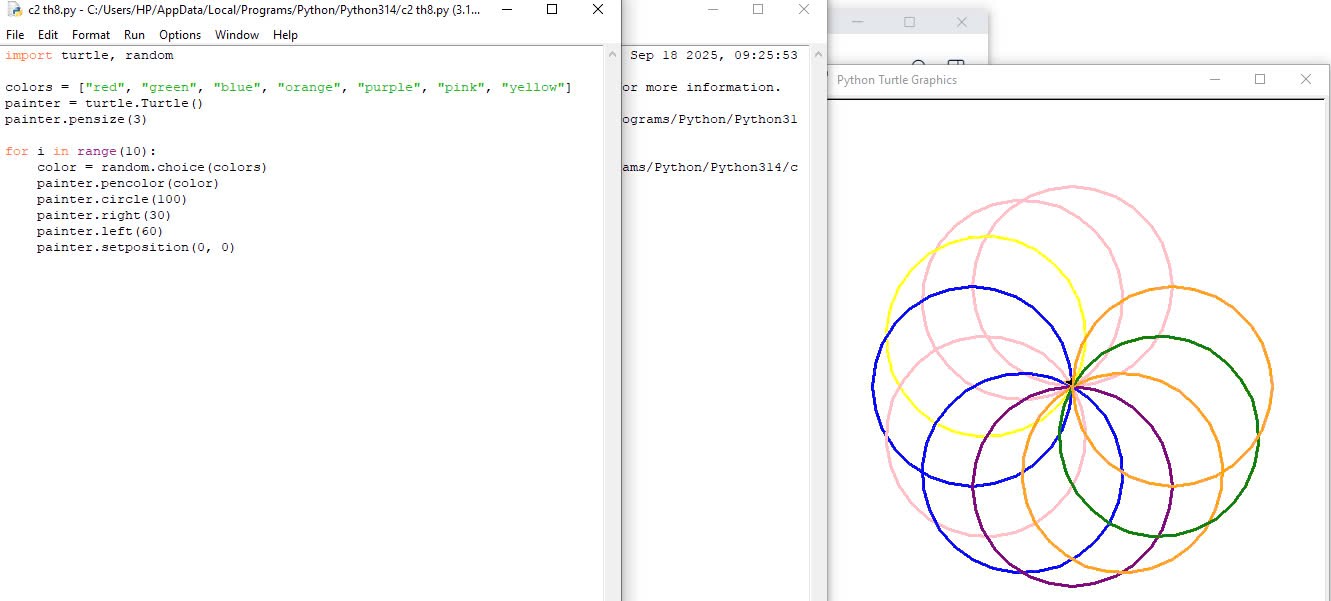
Painter.right(30)

Painter.left(60)

Painter.setposition(0, 0)

Turtle.done()

3.Dựa trên các kết quả đạt được từ các chương trình trên hãy viết chương trình hiển thị hình ảnh đồ họa sau:



Trương trình:  
 import turtle

import random

colors = ["red", "green", "blue", "orange", "purple", "pink", "yellow"]

p = turtle.Turtle()

p.speed(0)

p.pensize(3)

for i in range(12): # 12 vòng tròn

p.pencolor(random.choice(colors))

p.circle(100)

p.right(30) # mỗi vòng tròn xoay 30 độ

turtle.done()

4. Viết chương sử dụng thư viện đồ họa tkinter thực hiện:

* 1. Xây dựng cửa sổ đồ họa window form
  2. Thêm một widget (button) vào window form
  3. Xây dựng phương thức xử lý sự kiện phím bấm

From tkinter import \*

Window = Tk()

Window.title(“Demo Tkinter”)

Window.geometry(‘350x200’)

# Nhãn hiển thị

Lbl = Label(window, text=”Hello”, font=(“Arial”, 14))

Lbl.grid(column=0, row=0)

# Hàm xử lý sự kiện click

Def clicked():

Lbl.configure(text=”Button was clicked !!!”)

# Tạo nút bấm

Btn = Button(window,

Text=”Click Me”,

Command=clicked,

Bg=”lightblue”, # màu nền

Fg=”black”) # màu chữ

Btn.grid(column=1, row=0)

Window.mainloop()

5.Sử dụng thư viện tkinter thực hiện:

a)Xây dựng các radio button cho phép thực hiện các lựa chọn khác nhau

import tkinter as tk

from tkinter import messagebox

# Hàm được gọi khi giá trị của Radio Button thay đổi

Def show\_choice():

# Lấy giá trị hiện tại của biến selected\_option

Selected = selected\_option.get()

# Hiển thị thông báo với lựa chọn đã chọn

Messagebox.showinfo(“Lựa chọn”, f”Bạn đã chọn: {selected}”)

# Bạn có thể thêm các hành động khác ở đây tùy thuộc vào giá trị của selected

# 1. Tạo cửa sổ chính (main window)

Window = tk.Tk()

Window.title(“Thực hành Radio Button”)

Window.geometry(“400x200”)

# 2. Tạo biến điều khiển (IntVar) để theo dõi lựa chọn của Radio Button

# Giá trị của biến này sẽ thay đổi khi người dùng chọn một Radio Button khác

Selected\_option = tk.StringVar(value=”Lựa chọn A”) # Thiết lập giá trị mặc định

# 3. Tạo tiêu đề

Lbl\_title = tk.Label(window, text=”Vui lòng chọn một tùy chọn:”, font=(“Arial”, 12, “bold”))

Lbl\_title.pack(pady=10) # Dùng .pack() để bố trí đơn giản

# 4. Tạo các Radio Button

# Radio Button 1

# ‘text’ là nhãn hiển thị

# ‘variable’ liên kết với biến điều khiển

# ‘value’ là giá trị mà biến điều khiển sẽ nhận khi Radio Button này được chọn

# ‘command’ là hàm được gọi khi Radio Button này được chọn

Rad\_button\_a = tk.Radiobutton(

Window,

Text=”Lựa chọn A”,

Variable=selected\_option,

Value=”Lựa chọn A”,

Command=show\_choice

)

Rad\_button\_a.pack(anchor=tk.W) # anchor=tk.W căn lề trái

# Radio Button 2

Rad\_button\_b = tk.Radiobutton(

Window,

Text=”Lựa chọn B”,

Variable=selected\_option,

Value=”Lựa chọn B”,

Command=show\_choice

)

Rad\_button\_b.pack(anchor=tk.W)

# Radio Button 3

Rad\_button\_c = tk.Radiobutton(

Window,

Text=”Lựa chọn C”,

Variable=selected\_option,

Value=”Lựa chọn C”,

Command=show\_choice

)

Rad\_button\_c.pack(anchor=tk.W)

# 5. Chạy vòng lặp sự kiện chính

Window.mainloop()

* 1. **Câu hỏi kiểm tra**

- Giải thích chương trình đã thực hiện.

- Sử dụng các hàm trong thư viện turtle, tkinter để xây dựng các ứng dụng graphic

Với python.

* 1. **Tài liệu tham khảo**

[1]. Allen B. Downey, Think Python, O’Reilly Media, Inc, 2015. [2]. https://likegeeks.com/python-gui-examples-tkinter-tutorial/