|  |
| --- |
| **TRƯỜNG ĐẠI HỌC VINH**  **VIỆN KỸ THUẬT CÔNG NGHỆ** |



BÁO CÁO THỰC HÀNH

**KỸ THUẬT LẬP TRÌNH**

MÃ HỌC PHẦN: ELE20004

SVTH: NGUYỄN NHƯ CƯỜNG

MSSV: 235752020710047

GVHD: MAI THẾ ANH

**NGHỆ AN - 2025**

**MỤC LỤC**

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

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

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

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

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

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

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

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

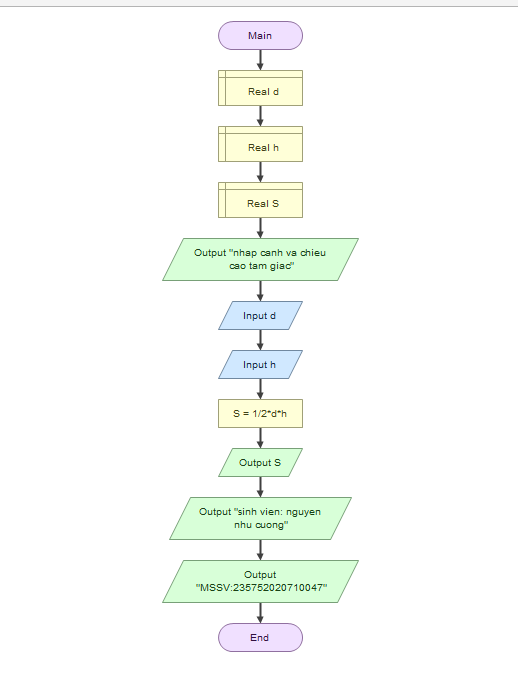
# ***Bài 1. Thực hiện các thuật toán bằng phần mềm Flowgorithm***

* 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
  1. **Các bước thực hiện và kết quả**

*(Tóm tắt các bước thực hiện và kết quả nhận được trong mỗi bước, giải thích theo yêu cầu, trình bày ngắn gọn đúng trọng tâm)*

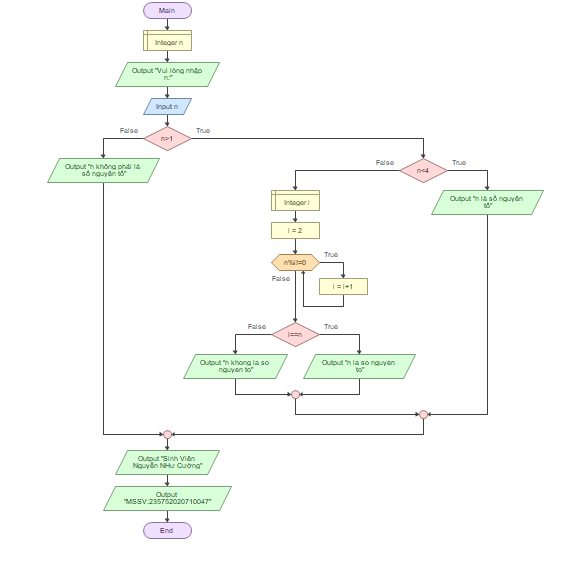
*(Có thể chụp màn hình nhưng phải bổ sung thêm khối in thông tin sinh viên như họ và tên, MSSV ra màn hình theo mẫu)*

Câu d. *Viết chương trình nhập và cạnh và chiều cao tương ứng của một tam giác và in ra .*



Kết quả:



Câu E. Xây dựng thuật toán kiểm tra tính nguyên tố (một số nguyên dương n là số nguyên tố khi chỉ có hai ước là 1 và chính nó)  


**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 câu lẹnh điều khiển khi viết một chương trình bằng ngôn ngữ python.

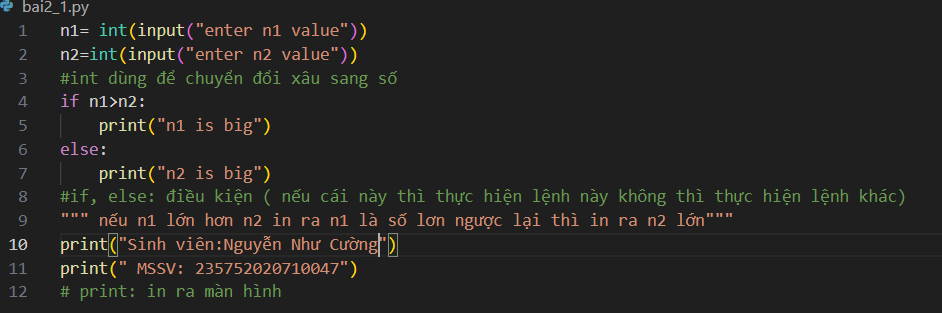
**2.2. Cơ sở lý thuyết**

Xem các khái niệm: Định danh, các từ khoá, dòng lệnh, khối lệnh, trích dẫn, comment, biến, toán tử, cấu trúc điều khiển trong python.

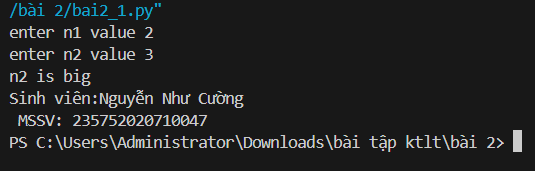
**2.3. Các bước tiến hành**

Các bài tập sử dụng bằng ngôn ngữ python:

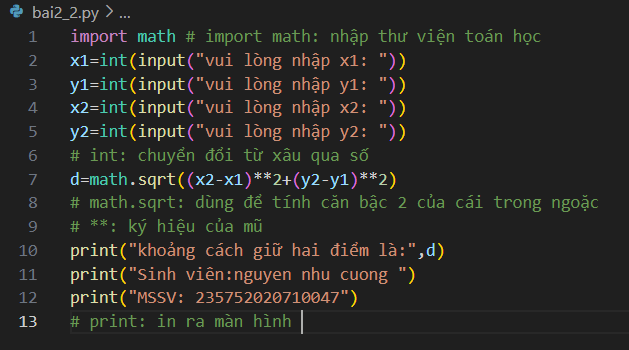
Câu 2.1 Viết đoạn chương trình sau và sửa lỗi:



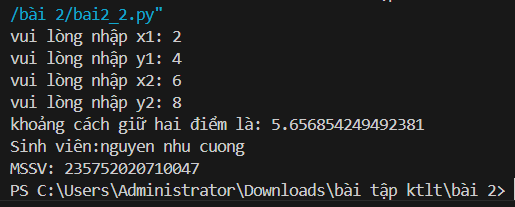
**Kết quả:**



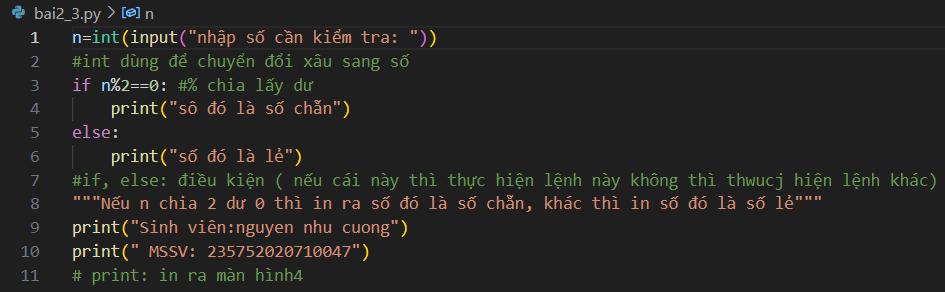
Câu 2.2. Viết chương trình nhập 2 điểm và tính khoảng cách:



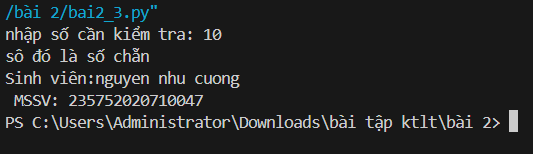
**Kết quả**:



Câu 2.3 Viết chương trình nhập vào một số vàkiểm tra số đó chẵn hay lẽ, in thông báo ra màn hình:



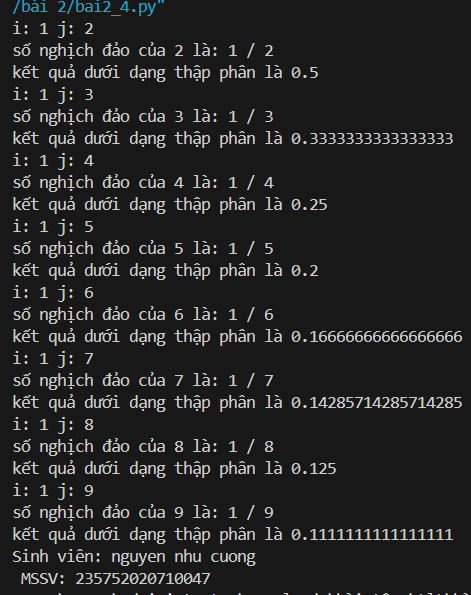
Kết quả:



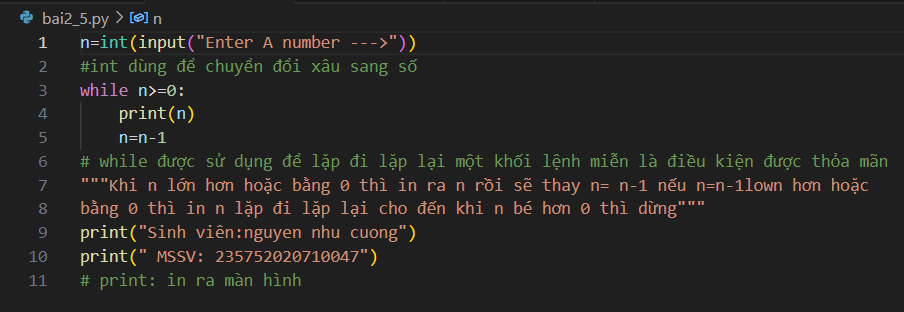
Câu 2.4. Viết chương trình in ra màn hình số nghịch đảo và kết quả dưới dạng thập phân của một dãy số tự nhiên trong khoảng (a,b)



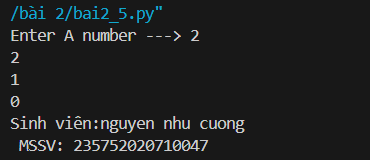
Kết quả:



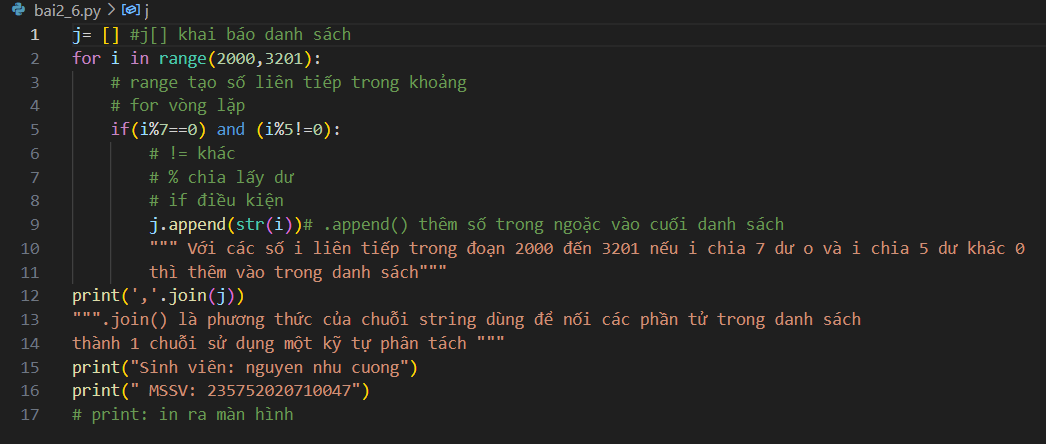
Câu 2.5. Viết chương trình nhạp vào một số tự nhiên n>0, in ra màn hình các số tự nhiên giảm dần từ n đến 0, mỗi ký tự in trên một hàng:



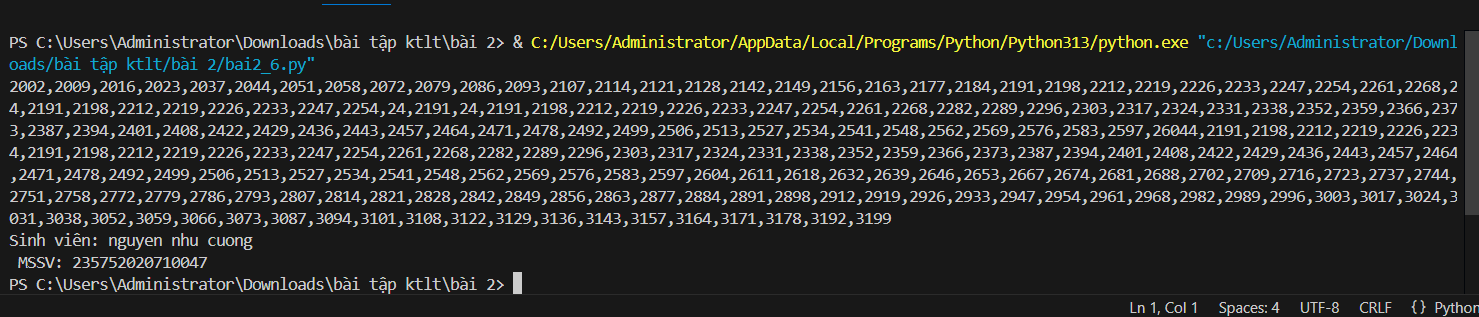
Kết quả:



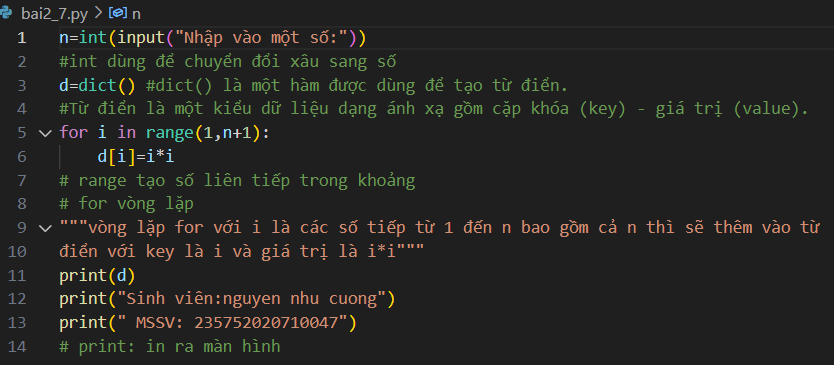
**Câu 2.6.** Viết chương trình tìm tất cả các số chia hết cho 7 nhưng không phải bội của số 5, nằm trong đoạn 2000 và 3200( tính cả 2000 và 3200). Các số thu được sẽ được in thành mỗi chuỗi trên một dòng, cách nhau bằng dấu phẩy:



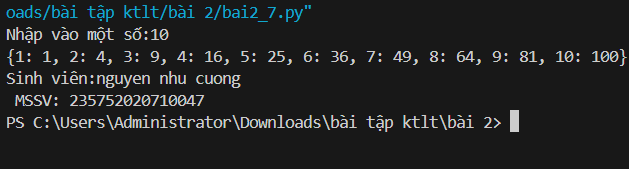
Kết quả:



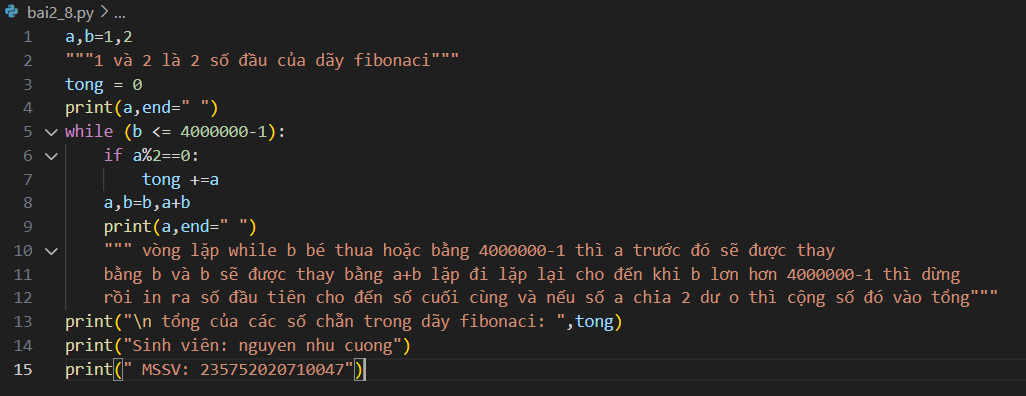
Câu 2.7. Với số nguyên n nhất định, hãy viết chương trình tạo ra một dictionary chứa( i, i\*i) như là số nguyên tử từ 1 đến ( bao gồm cả và và n) sau đó in ra dictionary này:



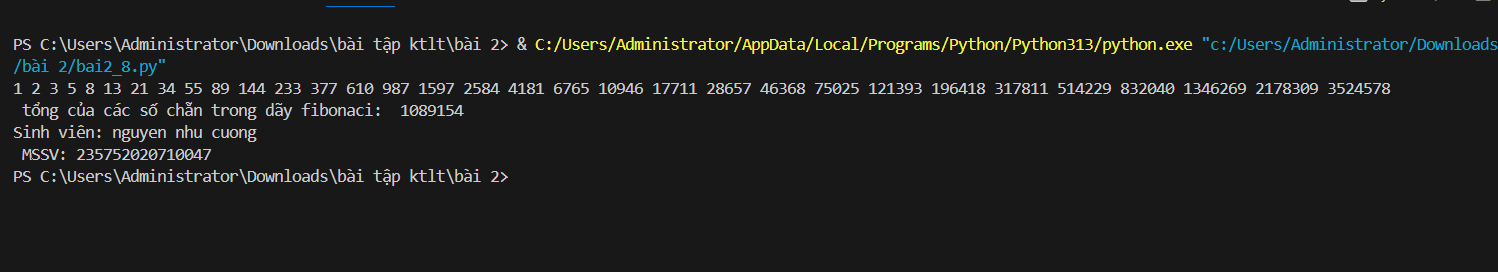
Kết quả:



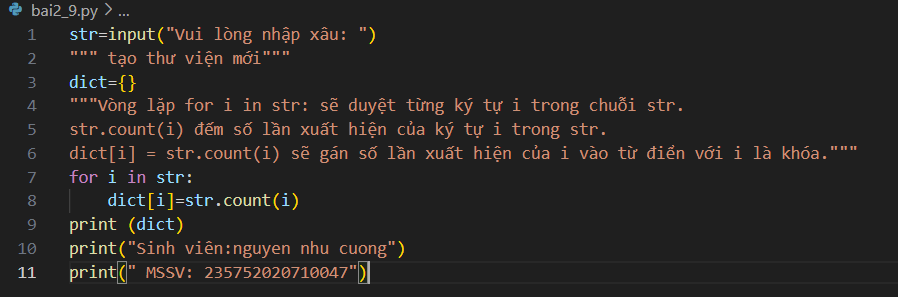
Câu 2.8. Viết chương trình in ra màn hình dãy số Fibonacci nhỏ hơn 4.000.000 tìm các số chẵn trong dãy đã in:



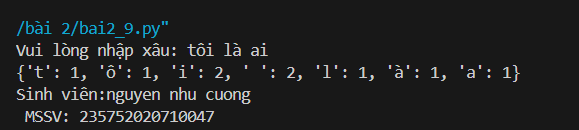
Kế quả:



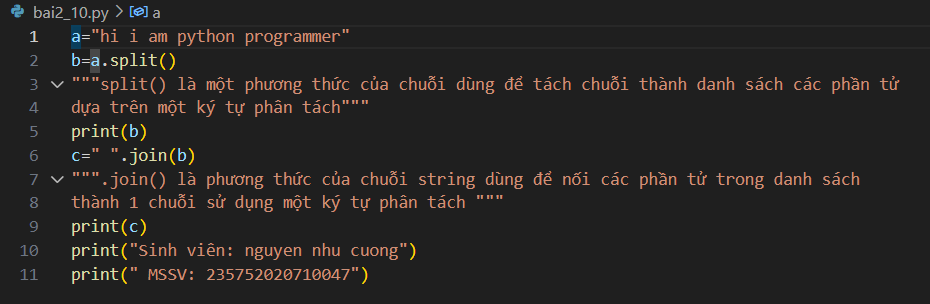
Câu 2.9. Viết chương trình đếm số ký tự trong một xâu ký tự nhập vào từ bàn phím, lưu các ký tự vào các cấu trúc từ điển:



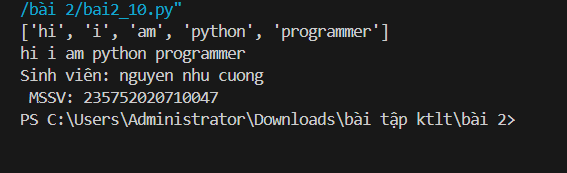
Kết quả:



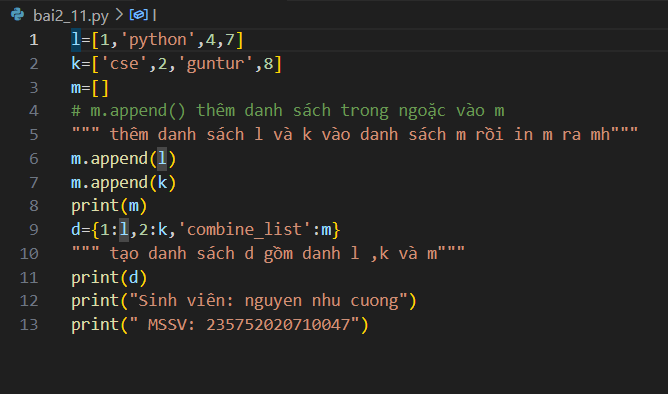
Câu 2.10. Viết chương trình sử sụng các phương thức split và join để tách nhập xâu ký tự:



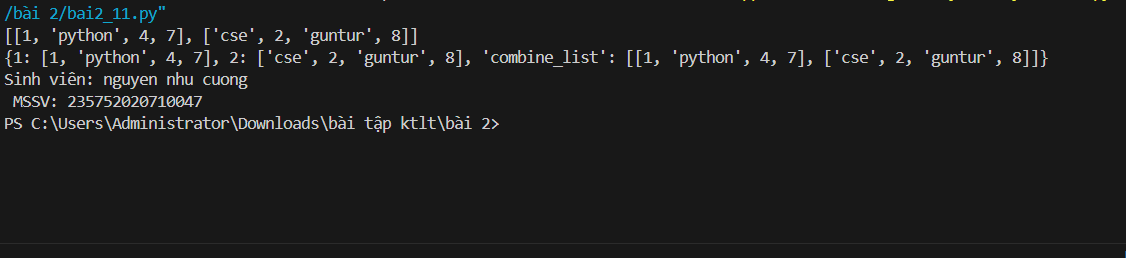
Kết quả:



Câu 2.11. Viết chương trình kết nối các danh sách vào từ điển:

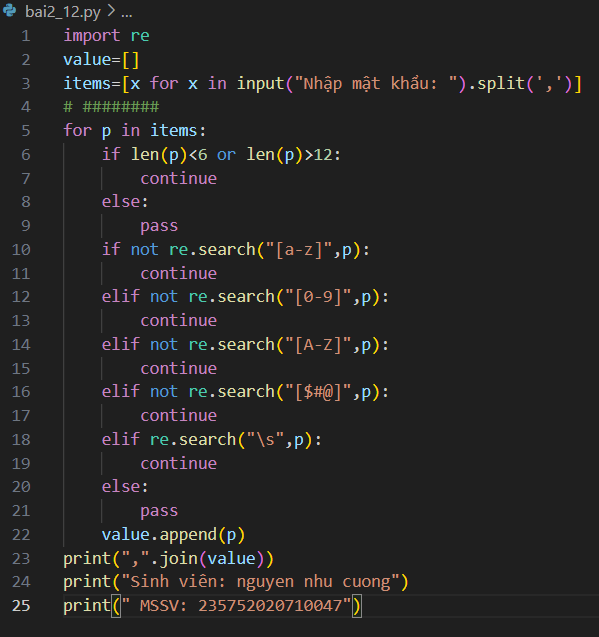


Kết quả:

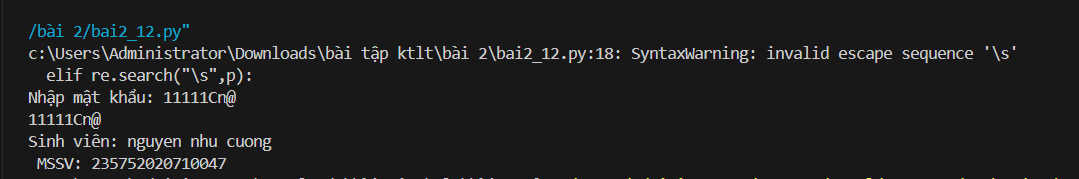


Câu 2.12. Một website yêu cầu người dùng nhập tên người dùng và mật khẩu để đăng ký. Viết một chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào.

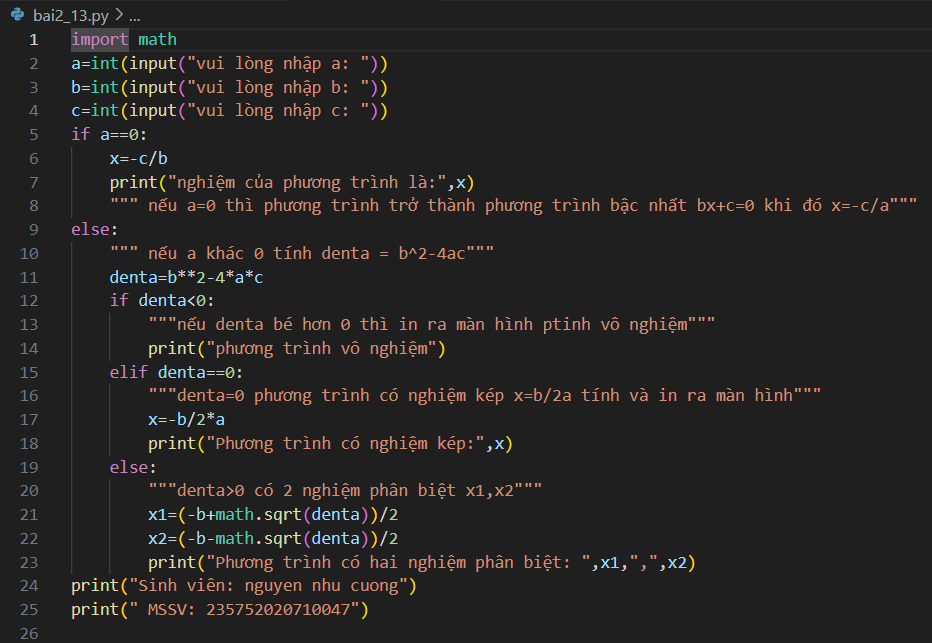
Các tiêu chí kiểm tra mật khẩu bao gồm:  
- Ít nhất 1 chữ cái nằm trong [a-z]  
- Ít nhất 1 số nằm trong [0-9]  
- Ít nhất 1 kí tự nằm trong [A-Z]  
- Ít nhất 1 ký tự nằm trong [$ # @]  
- Độ dài mật khẩu tối thiểu: 6  
- Độ dài mật khẩu tối đa: 12  
Chương trình phải chấp nhận một chuỗi mật khẩu phân 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.  
Ví dụ mật khẩu nhập vào chương trình là: ABd1234@1,a F1#,2w3E\*,2We3345  
Thì đầu ra sẽ là: ABd1234@1



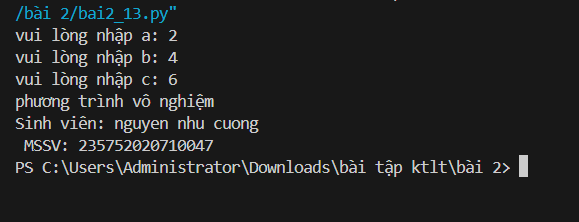
Kết quả:



Câu 2.13. Viết chương trình giải phương trình bậc 2: ax2+bx+c=0, với các hệ số a, b, c  
nhập từ bàn phím:



Kết quả:



## Câu hỏi kiểm tra

Các kiểu biến, khai bảo và đặt tên biến trong python.

Trong Python, có một số kiểu biến phổ biến như sau:

Kiểu số (Numeric types):

* int: Kiêu số nguyên, ví dụ: 1, 2, -3.
* float: Kiểu số thực, ví dụ: 3.14, -0.5.

Kiểu chuỗi (String):

* str: Kiều chuỗi, ví dụ: "Hello", "World", "123".

Kiểu Boolean (Boolean):

* bool: Kiểu logic, chỉ có hai giá trị là True hoặc False.

Kiêu danh sách (List):

* list: Kiều dữ liệu dạng danh sách, cho phép lưu trữ nhiều giá trị. Ví dụ: [1, 2, 3], ["apple", "banana", "orange"].

Kiểu tuple (Tuple):

* tuple: Kiểu dữ liệu dạng bộ, giống như danh sách nhưng không thể thay đổi giá trị sau khi khởi tạo. Vì dụ: (1, 2, 3), ("apple", "banana", "orange").

Kiểu từ điển (Dictionary):

* dict: Kiểu dữ liệu dạng từ điển, lưu trữ các cặp khóa giá trị. Ví dụ: {"name": "John", "age": 25).

2. Cấu trúc điều khiển trong Python.

a. Cấu trúc rẽ nhánh (Conditional statements):

* if: Sử dụng để thực thi một khối mã nếu một điều kiện được đáp ứng.
* if-else: Sử dụng để thực thi một khối mã nếu một điều kiện được đáp ứng và một khối mã khác nếu điều kiện không được đáp ứng.
* if-elif-else: Sử dụng để kiểm tra nhiều điều kiện khác nhau và thực thi các khối mà tương ứng.

b. Cấu trúc lập (Looping statements):

* for: Sử dụng để lặp qua một chuỗi giá trị hoặc một đối tượng có thể lặp.
* while: Sử dụng để lặp lại một khối mã cho đến khi một điều kiện được đáp ứng

c. Cấu trúc điều khiển rời rạc (Branching statements):

* break: Sử dụng để thoát khỏi vòng lập hiện tại.
* continue: Sử dụng để bỏ qua phần còn lại của vòng lặp hiện tại và chuyển đến lần lập tiếp theo.
* pass: Sử dụng để không làm gì trong một khối mã hoặc hàm, chỉ để đảm bảo cú pháp hợp lệ.

**Bài 3: Lập trình hàm trong Python**

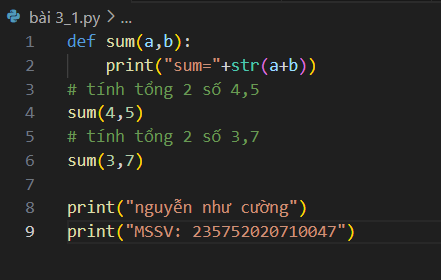
**3.1. Mục đích**

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

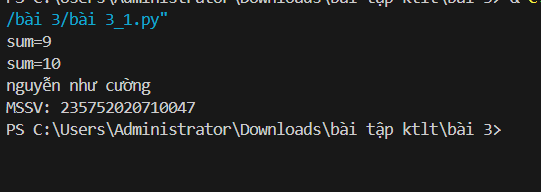
**3.2. Cơ sở lý thuyết**

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.

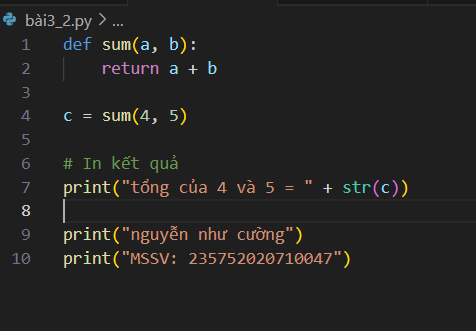
**3.3. Các bước tiến hành**Câu 3.1. Viết hàm sum() tính tổng hai số:

****

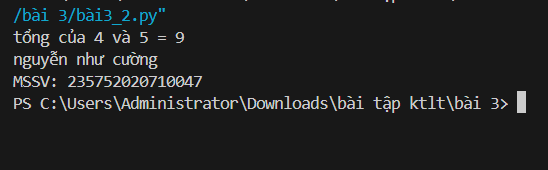
Kết quả:



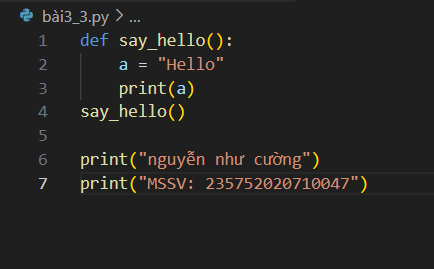
Câu 3.2. Viết hàm sum() với kết quả trả về:



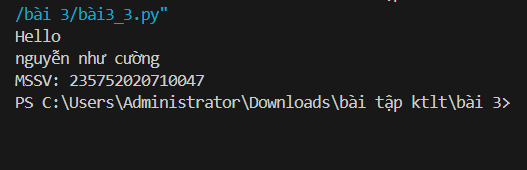
Kết quả:



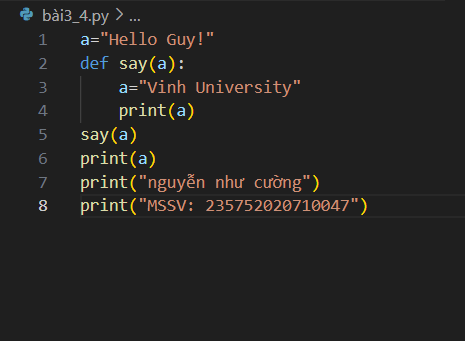
Câu 3.3. Tìm và sửa lỗi chương trình:



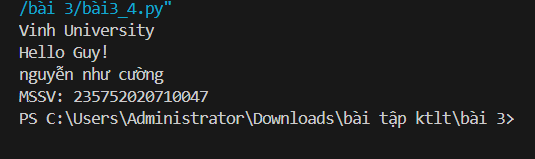
Kết quả:



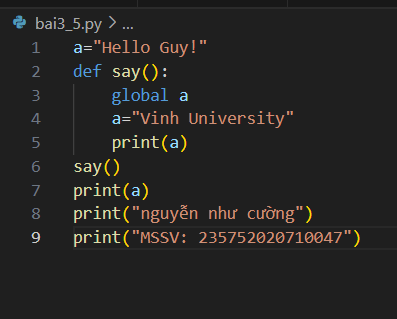
Câu 3.4. Viết chương trình có phạm vi biến như sau:



Kết quả:



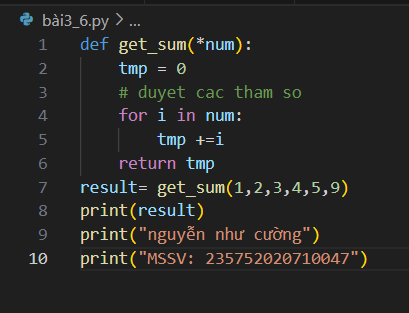
Câu 3.5. Viết chương trình sau và xem sự thay đổi của biến:



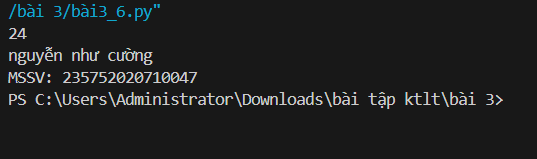
Kết quả:



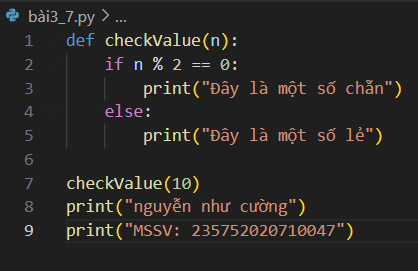
Câu 3.6. Viết chương trình sau và giải thích việc truyền tham số của hàm:



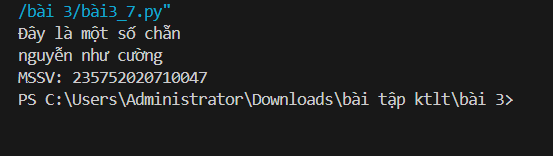
Kết quả:



Câu 3.7. 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.  
Gợi ý:  
Sử dụng toán tử % để kiểm tra xem số đó chẵn hay lẻ



Kết quả:

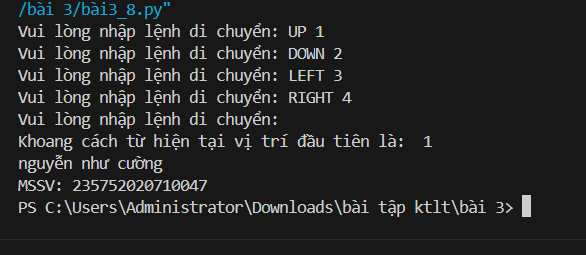


Câu 3.8. Một Robot di chuyển trong mặt phẳng bắt đầu từ điểm đầu tiên (0,0). Robot có thể di chuyển theo hướng UP, DOWN, LEFT và RIGHT với những bước nhất định.  
Dấu di chuyển của robot được đánh hiển thị như sau:  
UP 5  
DOWN 3  
LEFT 3  
RIGHT 3

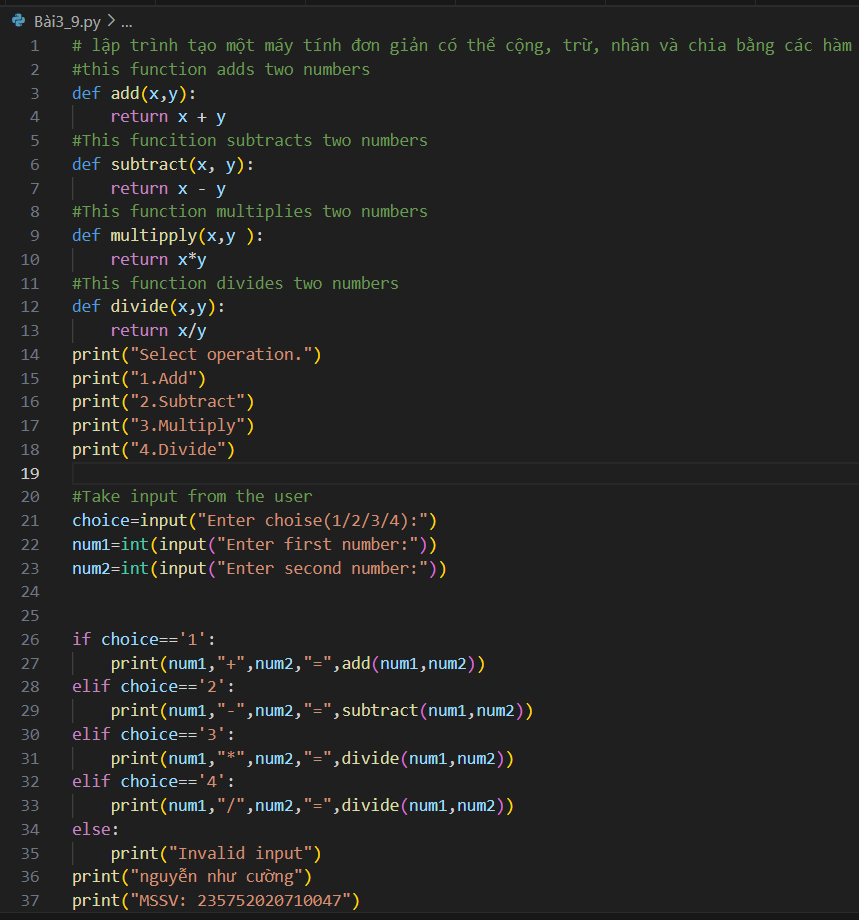
Các con số sau phía sau hướng di chuyển chính là số bước đi. Hãy viết chương  
trình để tính toán khoảng cách từ vị trí hiện tại đến vị trí đầu tiên, sau khi robot đã di chuyển một quãng đường. Nếu khoảng cách là một số thập phân chỉ cần in số nguyên gần nhất.



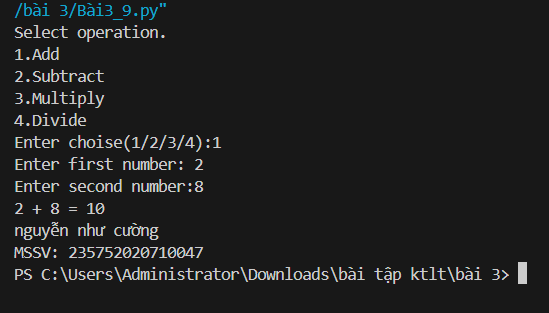
Kết quả:



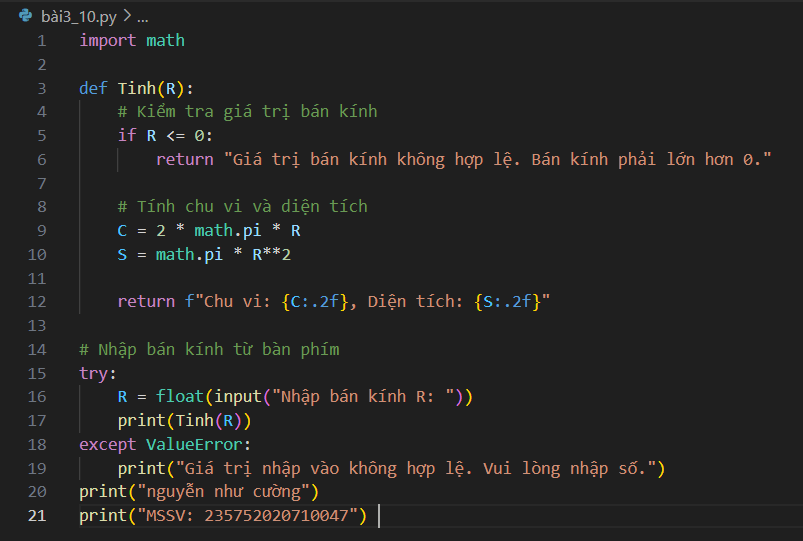
Câu 3.9. Chương trình máy tính thực hiện các phép tính đơn giản:



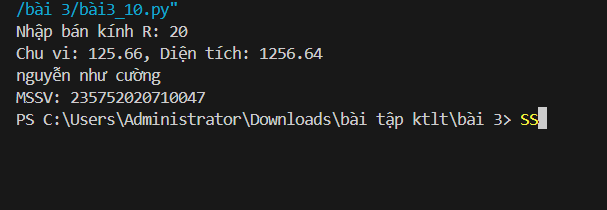
Kết quả:



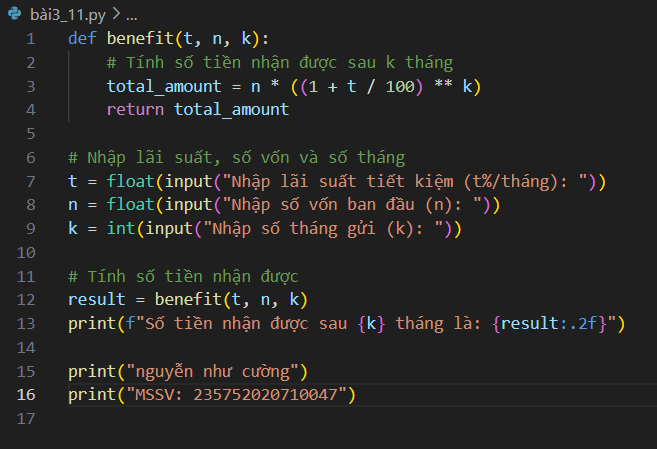
Câu 3.10. . Viết hàm “def Tinh(R):” tính chu vi và diện tích hình tròn, với bán kính R được nhập từ bàn phím, và kiểm tra giá trị bán kính đầu vào là hợp lệ.  
Gợi ý: sử dụng thư viện “import math” và math.pi cho số pi nếu cần:



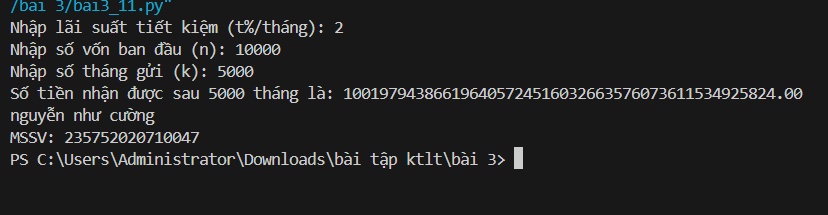
Kết quả:



Câu 3.11. Biết lãi suất tiết kiệm là t%/tháng (nhập t từ bàn phím). Nhập số vốn ban đầu n và số tháng gửi k. Tính số tiền nhận được sau k tháng sử dụng cấu trúc hàm def benefit(t,n,k):



Kết quả:



1.5.Câu hỏi kiểm tra

Giải thích 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 (Function declaration): Khai báo hàm trong Python sử dụng từ khóa "def" theo cú pháp sau:

def ten ham(tham sol, tham so2, ...):

# Mã lệnh trong hàm

…

Trong đó, "ten ham" là tên của hàm, "tham sol", "tham so2",... là các tham số truyền vào hàm (có thể không có tham số), và mà lệnh trong hàm được viết trong khối lệnh bên trong.

b. Gọi hàm (Function call): Để gọi một hàm trong Python, , ta sử dụng tên hàm kèm theo các đổi số (nếu có) được truyền vào đông trong dầu ngoặc đơn.

Ví dụ:

ten\_ham(gia\_tril, gia\_tri2, ...)

Trong đó, "ten\_ham" là tên của hảm cần gọi, "gia\_tril", "gia\_trì2", ... là các giá trị được truyền vào cho các tham số tương ứng.

c. Giá trị trả về (Return value): Một hàm trong Python có thể trà về một giá trị bằng từ khóa "return".

Ví dụ:

def tinh\_tong(a, b):

tong = a+b

return tong

Trong vì dụ trên, hàm "tỉnh tong" nhận hai tham số a và b, tỉnh tổng của chúng và trả về giả trị tổng đó. Khi gọi hàm và nhận giá trị trả về, ta có thể gán giá trị đó cho một biển hoặc sử dụng trực tiếp.

d. Tham số truyền vào (Function parameters): Tham số là các biến được khai báo trong khai báo hàm và được sử dụng trong mà lệnh của hàm. Khi gọi hàm, ta truyền giá trị cho các tham số này.

Vì dụ:

def chao(mo ta):

print("Xin chào.", mo ta)

chao("thế giới")

Trong ví dụ trên, hám "chao" có một tham số "mo\_ta". Khi gọi hàm "chao" và truyền giá trị "thế giới", hàm sẽ in ra "Xin chào, thế giới".

##### e. Tham số mặc định (Default parameters): Trong khai bảo hàm, ta có thể định nghĩa giá trị mặc định cho các tham số. Giá trị mặc định sẽ được sử dụng khi không có giá trị được truyền cho tham số đó khi gọi hàm.

##### Ví dụ:

##### def cong(a, b)):

##### return a + b

##### print(cong(3)) # Kết quả: 3

##### print(cong(3.5)) # Kết quả: 8

##### Trong ví dụ trên, hàm "cong" có một tham số "b" với giá trị mặc định là 0. Khi gọi hàm, nếu không truyền giá trị cho tham số "b", giá trị mặc định 0 sẽ được sử dụng.

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

**4.1. Mục đích**

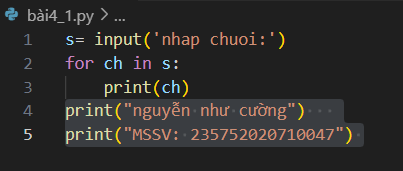
Giúp sinh viên nắm bắt các kiểu dữ liệu có cấu trúc trong lập trình python.

**4.2. Cơ sở lý thuyết**

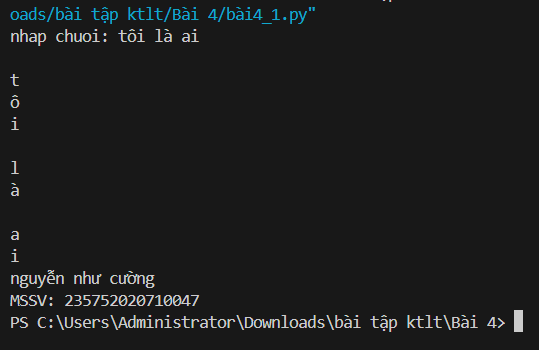
Xem các quy tắc sử dụng các kiểu dữ liệu: chuỗi, số, list, tuple, set và dictionary trong python.

**4.3. Các bước tiến hành**

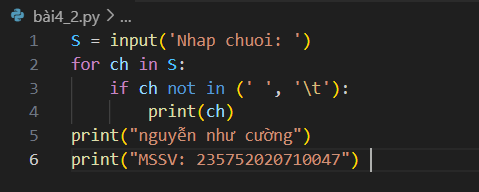
Câu 4.1. Nhập chuỗi S và in ra từng kí tự của S, mỗi kí tự trên một dòng.



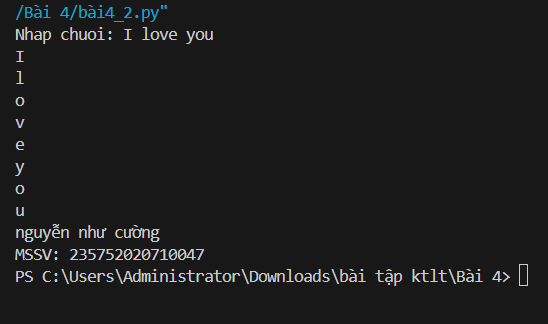
Kết quả:



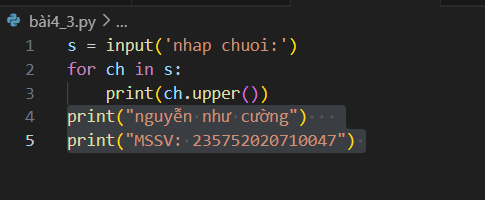
Câu 4.2. Chỉnh sửa ví dụ trên: hãy bỏ qua không in ra những kí tự “không nhìn thấy” (dấu space và dấu tab):



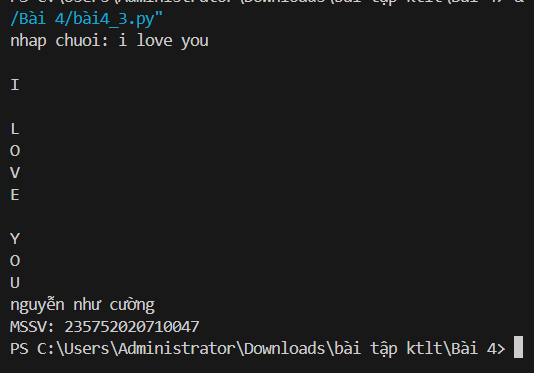
Kết quả:



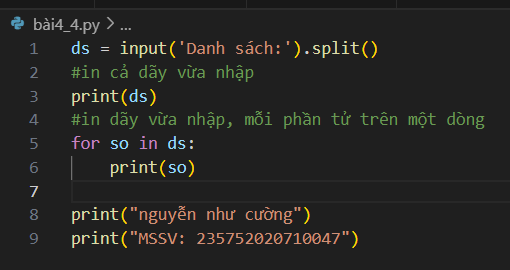
Câu 4.3. Chỉnh sửa ví dụ ở bài 1: hãy các kí tự ở dạng IN HOA:



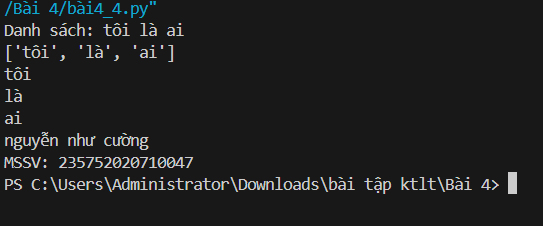
Kết quả:



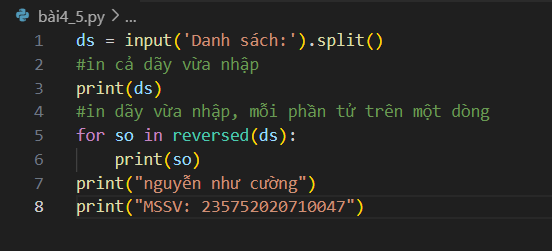
Câu 4.4. Nhập một danh sách trên một dòng, mỗi phần tử cách nhau bởi dấu trống hoặc tab, sau đó in ra dãy vừa nhập ra màn hình:



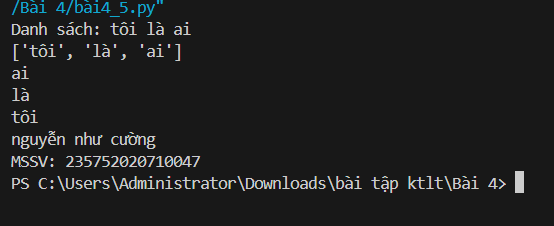
Kết quả:

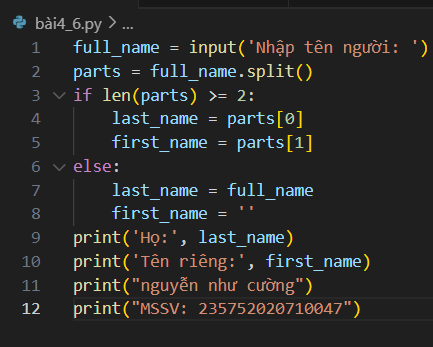


Câu 4.5. Chỉnh sửa ví dụ ở bài 4: nhập 1 danh sách các từ từ bàn phím, in ra các từ đó theo thứ tự ngược lại thứ tự vừa nhập (ví dụ nhập “mot hai ba” thì in ra “ba hai mot”)

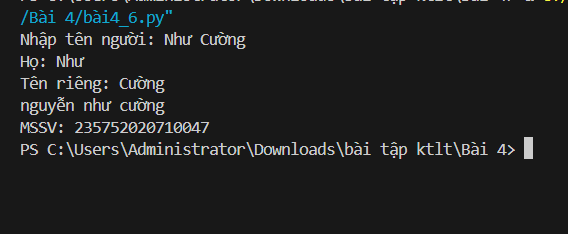


Kết quả:

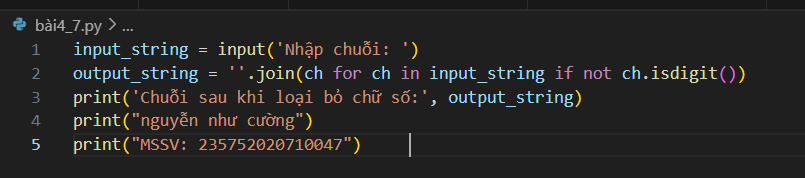


Câu 4.6. Nhập một tên người từ bàn phím, hãy tách phần họ và tên riêng của người đó và in chúng ra màn hình (giả thiết họ và tên riêng chỉ gồm một âm)  


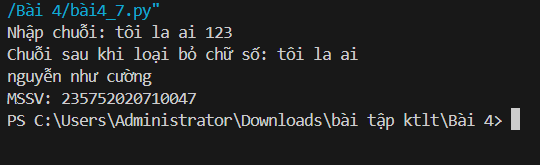
Kết quả:



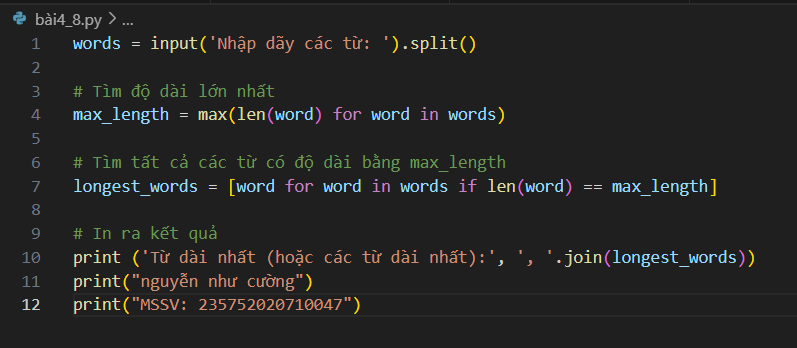
Câu 4.7. Nhập một chuỗi từ bàn phím, hãy loại bỏ tất cả các chữ số khỏi chuỗi và in lại nội dung chuỗi mới ra màn hình.



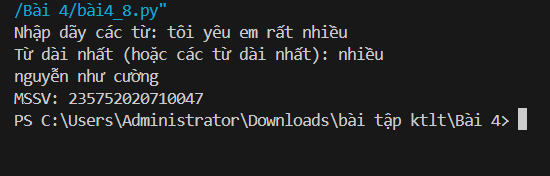
Kết quả:



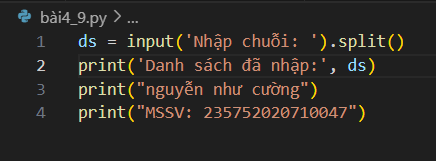
Câu 4.8. Nhập một dãy các từ từ bàn phím, hãy in ra từ dài nhất trong dãy vừa nhập, in ra mọi từ có cùng độ dài nhất.



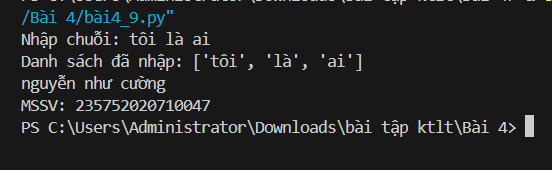
kết quả:



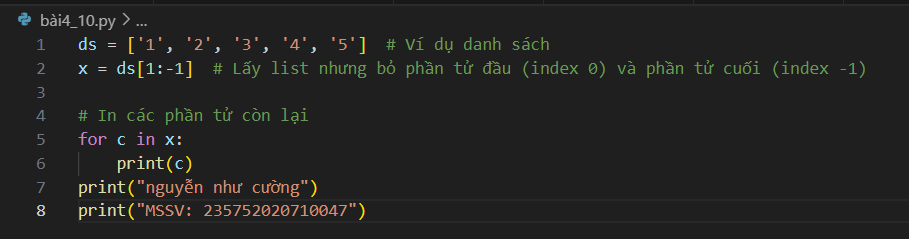
Câu 4.9. Nhập một list từ bàn phím:



Kết quả:



Câu 4.10. Cắt list: lấy list nhưng bỏ phần tử đầu và cuối:

****

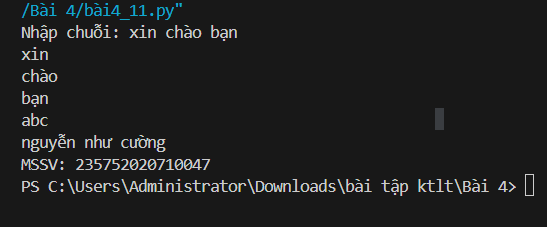
Kết quả:



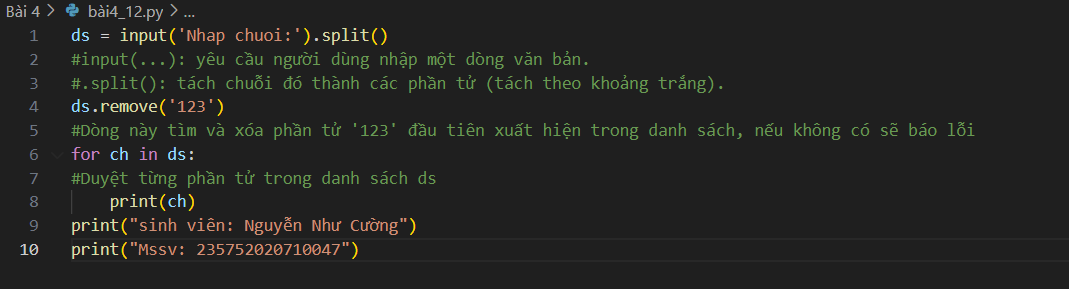
Câu 4.11. Thêm phần tử vào list:



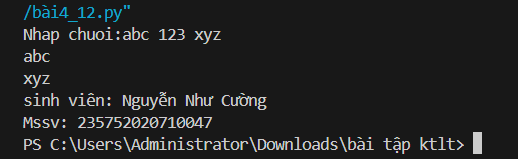
Kết quả:



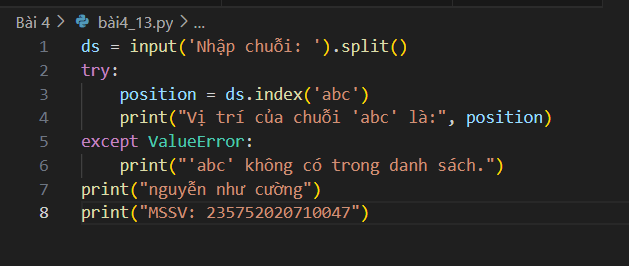
Câu 4.12. Bỏ phần tử khỏi list:



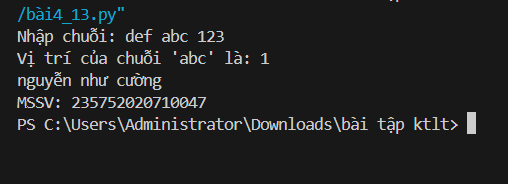
Kết quả:



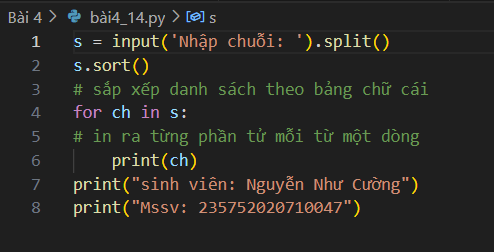
Câu 4.13. Tìm kiếm phần tử trong list



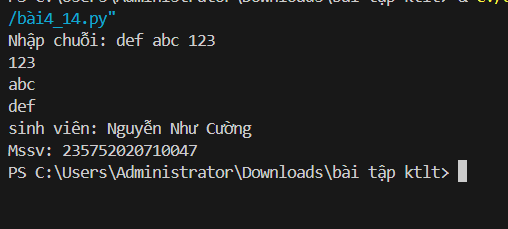
Kết quả:



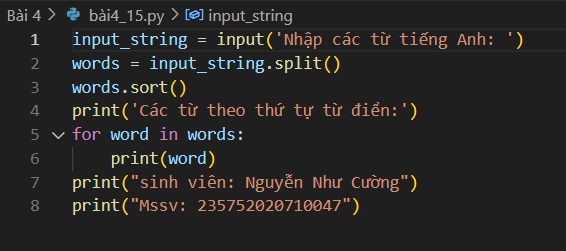
Câu 4.14. Sắp xếp các phần tử trong list:



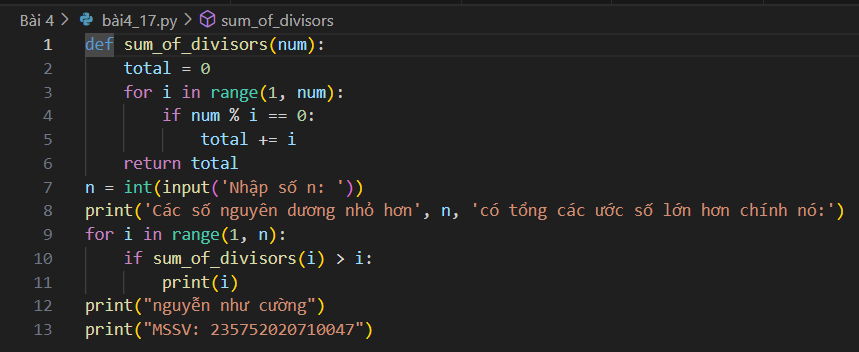
Kết quả:



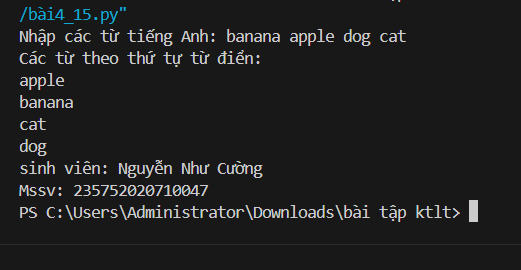
Câu 4.15. Người dùng nhập từ bàn phím liên tiếp các từ tiếng Anh viết tách nhau bởi dấu cách. Hãy nhập chuỗi đầu vào và tách thành các từ sau đó in ra màn hình các từ đó theo thứ tự từ điển.



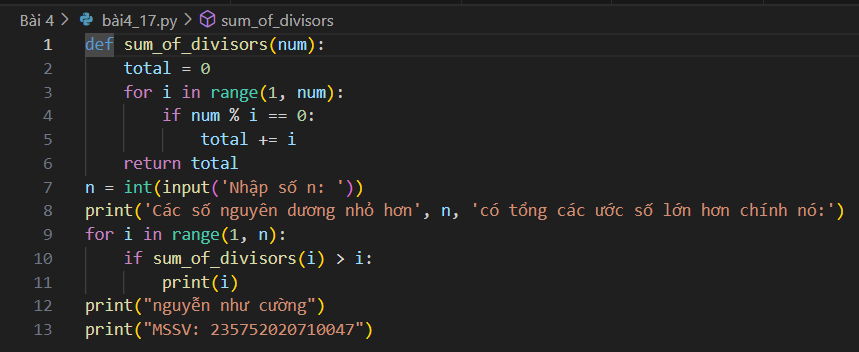
Câu 4.17. Nhập số n, in ra màn hình các số nguyên dương nhỏ hơn n có tổng các ước số lớn hơn chính nó.



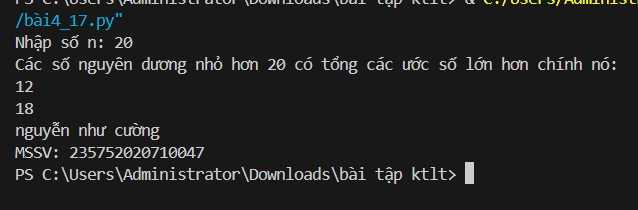
Kết quả:



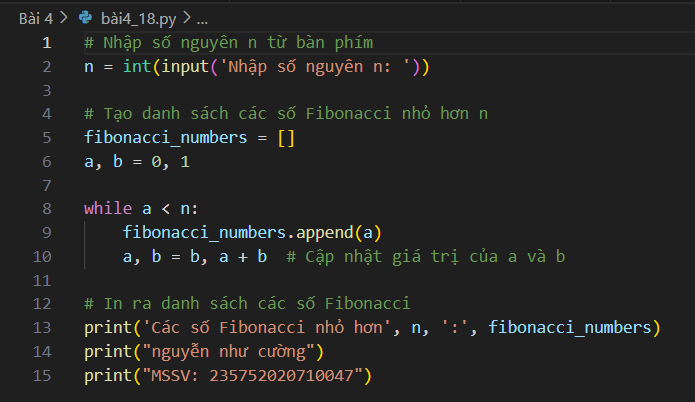
Câu 4.16. Người dùng nhập từ bàn phím chuỗi các số nhị phân viết liên tiếp được nối nhau bởi dấu phẩy. Hãy nhập chuỗi đầu vào sau đó in ra những giá trị được nhập.



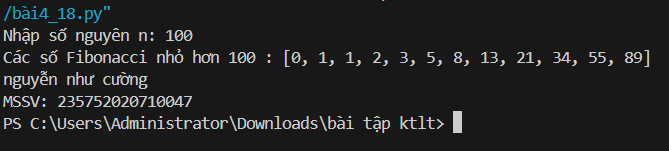
Kết quả:



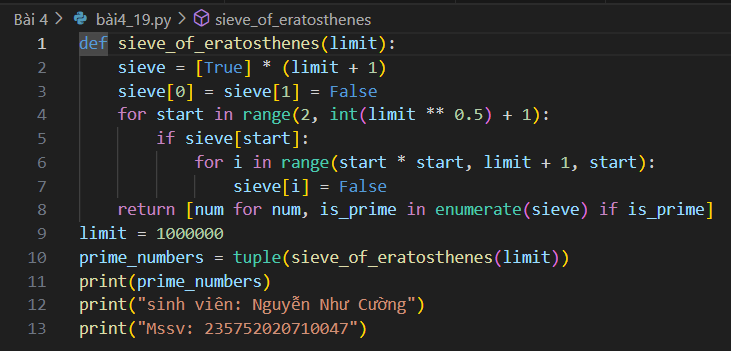
Câu 4.18. . Hãy nhập số nguyên n, tạo một list gồm các số fibonacci nhỏ hơn n và in ra màn hình.



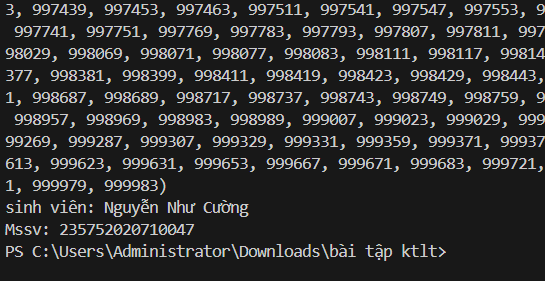
Kết quả:



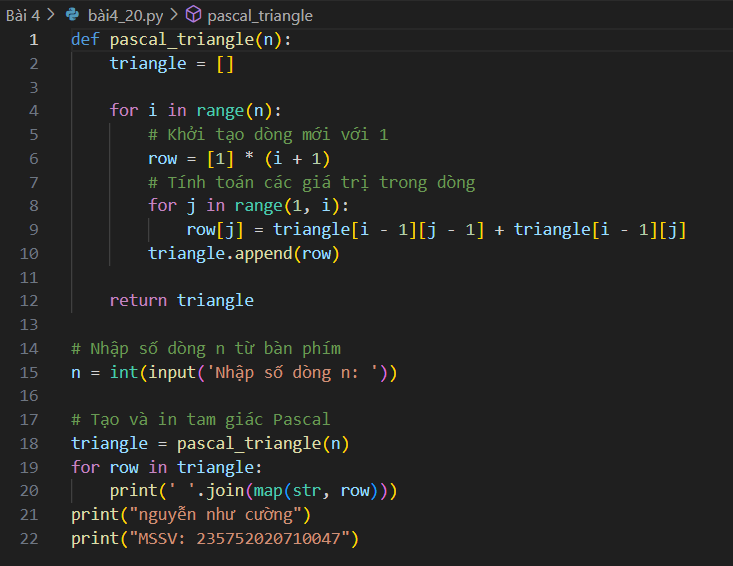
Câu 4.19. . Hãy tạo ra tuple P gồm các số nguyên tố nhỏ hơn 1 triệu.



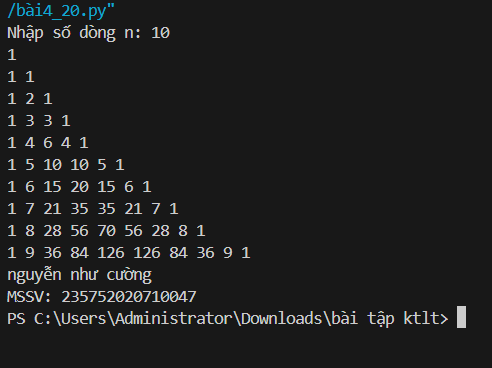
Kết quả:



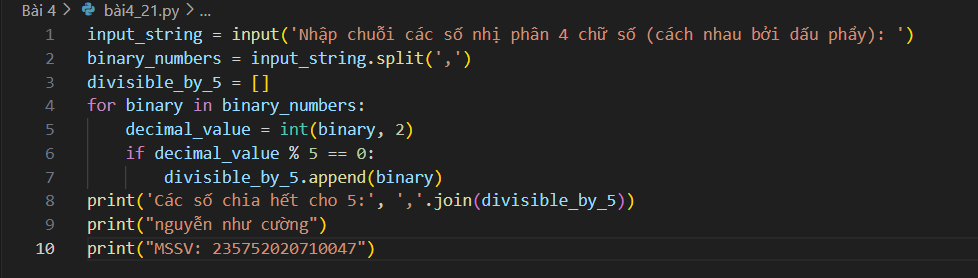
Câu 4.20. Nhập n, in n dòng đầu tiên của tam giác pascal.



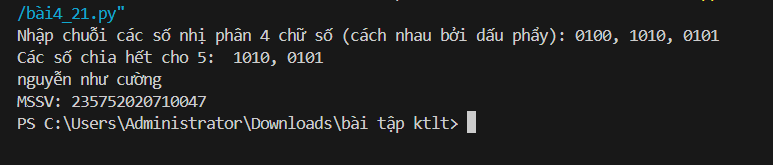
Kết quả:



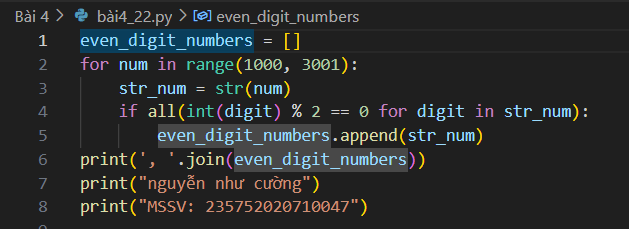
Câu 4.21. Viết một chương trình chấp nhận đầu vào là chuỗi các số nhị phân 4 chữ số, phân tách bởi dấu phẩy, kiểm tra xem chúng có chia hết cho 5 không. Sau đó in các số chia hết cho 5 thành dãy phân tách bởi dấu phẩy:



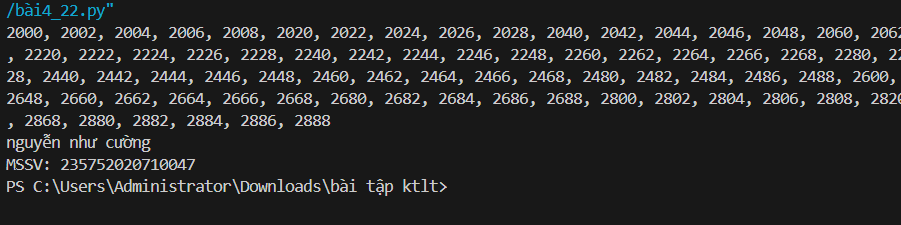
Kết quả:



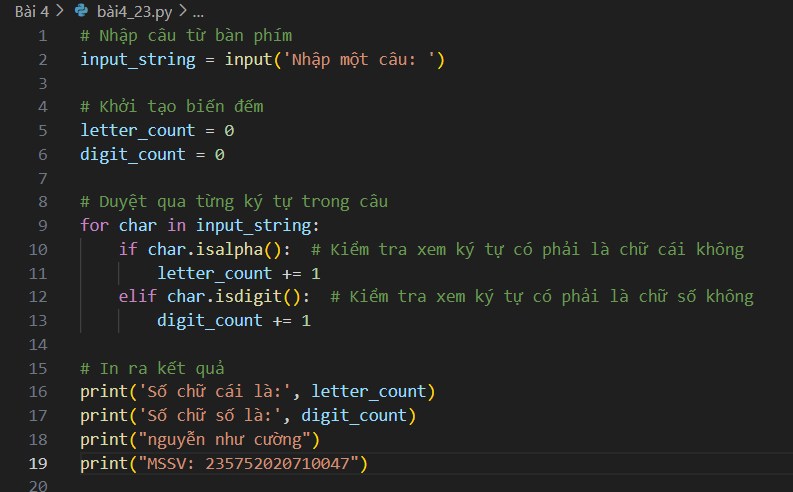
Câu 4.22. Viết một chương trình tìm tất cả các số trong đoạn 1000 và 3000 (tính cả 2 số này) sao cho tất cả các chữ số trong số đó là số chẵn. In các số tìm được thành chuỗi cách nhau bởi dấu phẩy, trên một dòng:



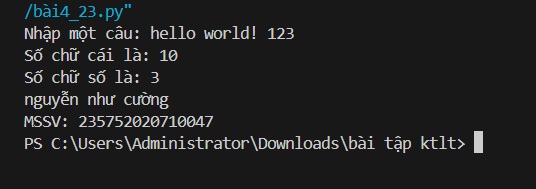
Kết quả:



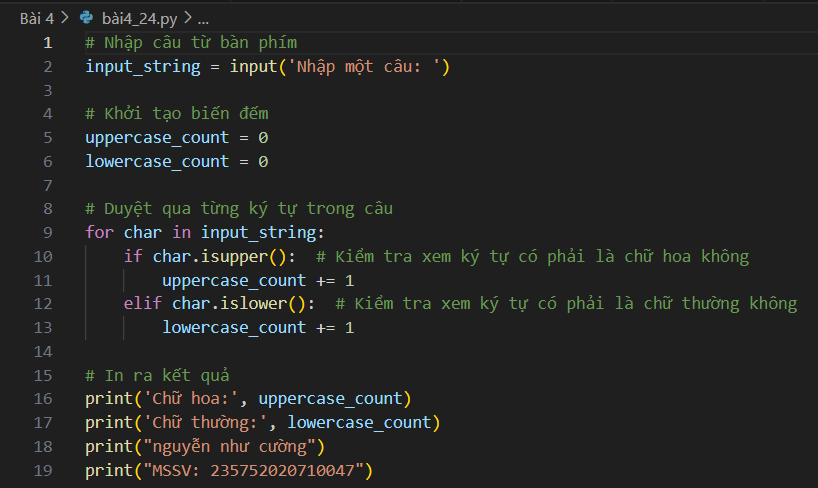
Câu 4.23. Viết một chương trình chấp nhận đầu vào là một câu, đếm số chữ cái và chữ số trong câu đó. Giả sử đầu vào sau được cấp cho chương trình: hello world! 123 Thì đầu ra sẽ là:  
Số chữ cái là: 10  
Số chữ số là: 3



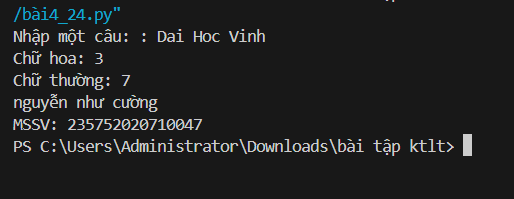
Kết quả:



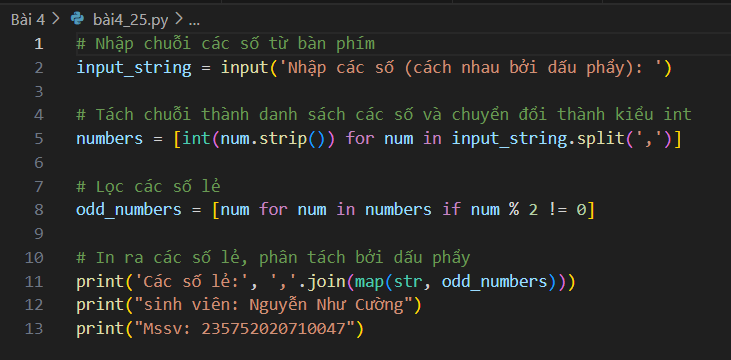
Câu 4.24. Viết một chương trình chấp nhận đầu vào là một câu, đếm chữ hoa, chữ thường. Giả sử đầu vào là: Dai Hoc Vinh Thì đầu ra là:  
Chữ hoa: 3  
Chữ thường: 7



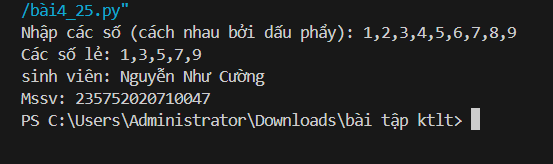
Kết quả:



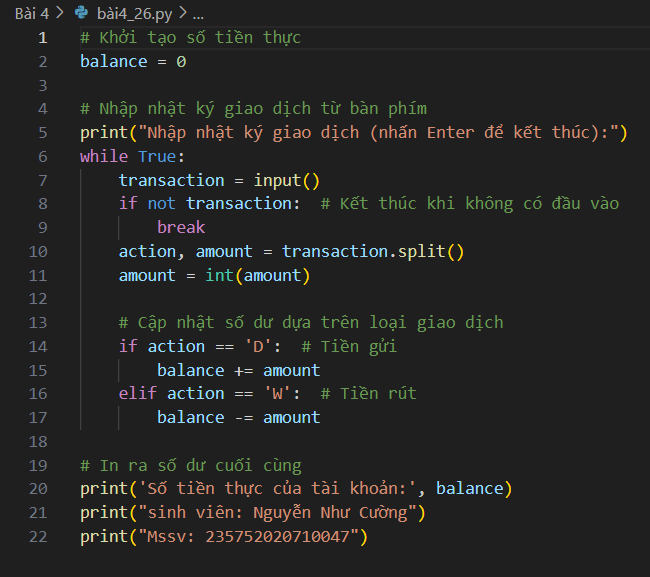
Câu 4.25. Sử dụng một danh sách để lọc các số lẻ từ danh sách được người dùng nhập vào. Giả sử đầu vào là: 1,2,3,4,5,6,7,8,9 thì đầu ra phải là: 1,3,5,7,9:



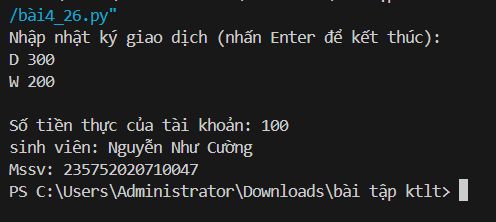
Kết quả:



Câu 4.26. Viết chương trình tính số tiền thực của một tài khoản ngân hàng dựa trên nhật ký giao dịch được nhập vào từ giao diện điều khiển.  
Định dạng nhật ký được hiển thị như sau:  
D 100  
W 200  
(D là tiền gửi, W là tiền rút ra).  
Giả sử đầu vào được cung cấp là:  
D 300  
D 300  
W 200  
D 100  
Thì đầu ra sẽ là:  
500



Kết quả:



**Bài 5: Thiết kế module trong Python**

5.1. Mục đích

Giúp sinh viên nắm bắt việc thiết kế module trong lập trình python, sử dụng

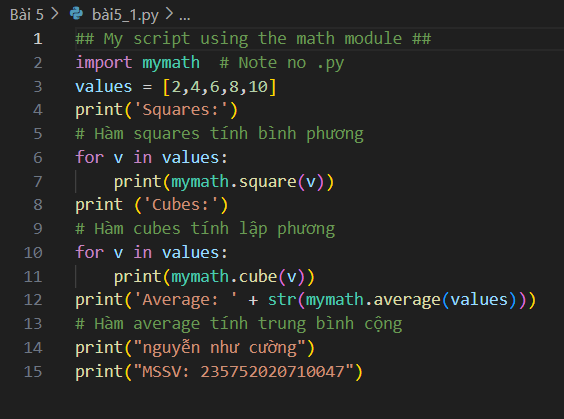
module thue viện numpy trong các ứng dụng.

5.2. **Cơ sở lý thuyết**

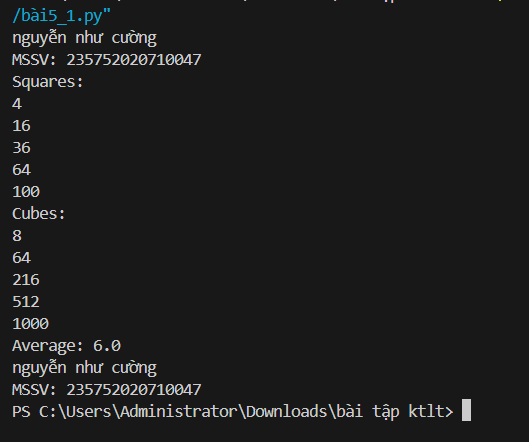
Xem các quy tắc khai báo, thiết kế và sử dụng module trong python, các thuật toán tìm kiếm, sắp xếp, cài đặt và sử dụng thư viện numpy.

5.3. **Các bước tiến hành**

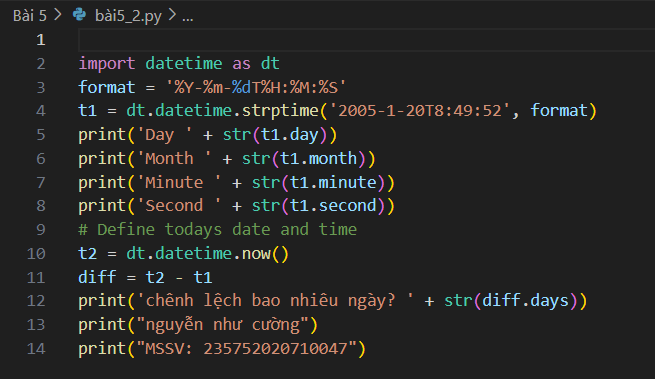
**Câu 5.1. Sử dụng module. Định nghĩa một module toán học gọi là mymath và sử dụng module này từ một tập lệnh riêng biệt**



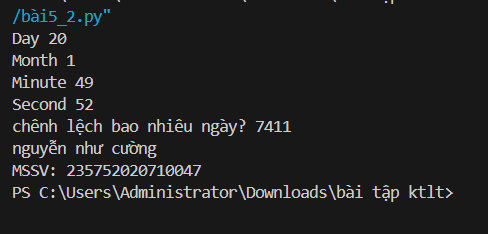
Kết quả:



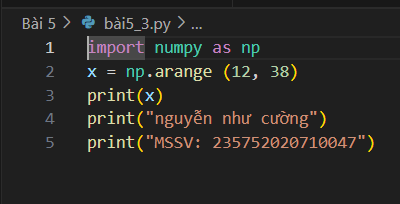
Câu 5.2. Sử dụng thư viện tiêu chuẩn của python (datetime)



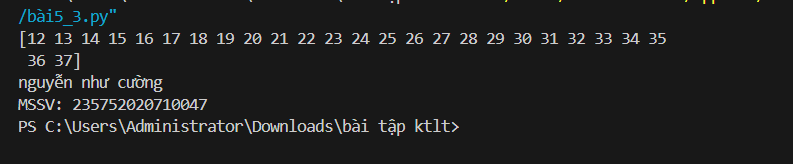
Kết quả:



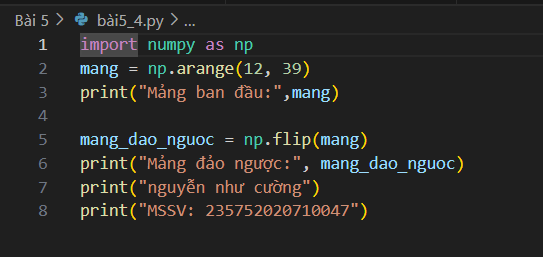
Câu 5.3. Viết chương trình sử dụng thư viện NumPy để tạo một mảng với các giá trị nằm trong khoảng từ 12 đến 38



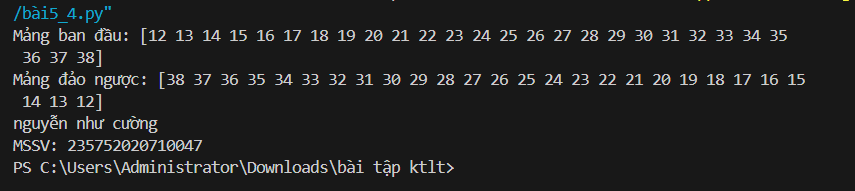
Kết quả:



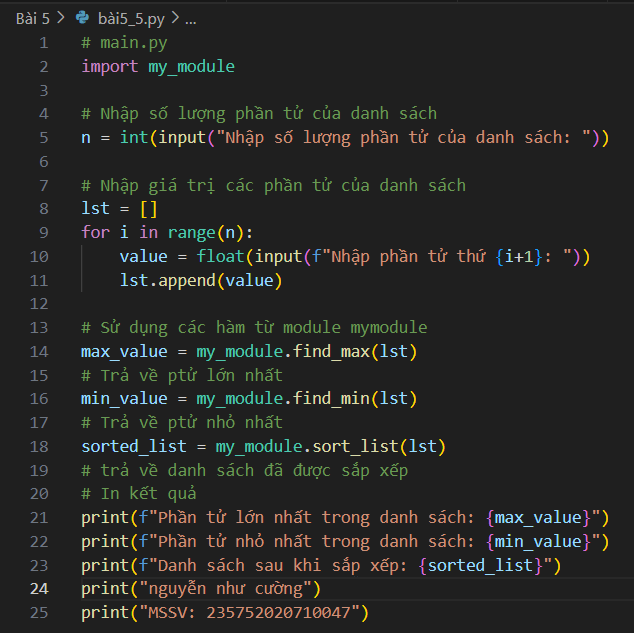
Câu 5.4. Viết chương trình để tạo một mảng với các giá trị nằm trong khoảng từ 12 đến 38 và đảo ngược mảng đã tạo (phần tử đầu tiên trở thành cuối cùng)



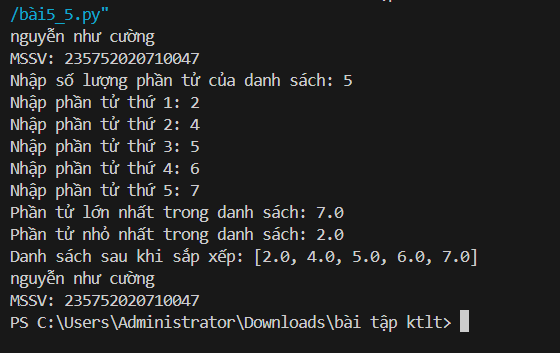
Kết quả:



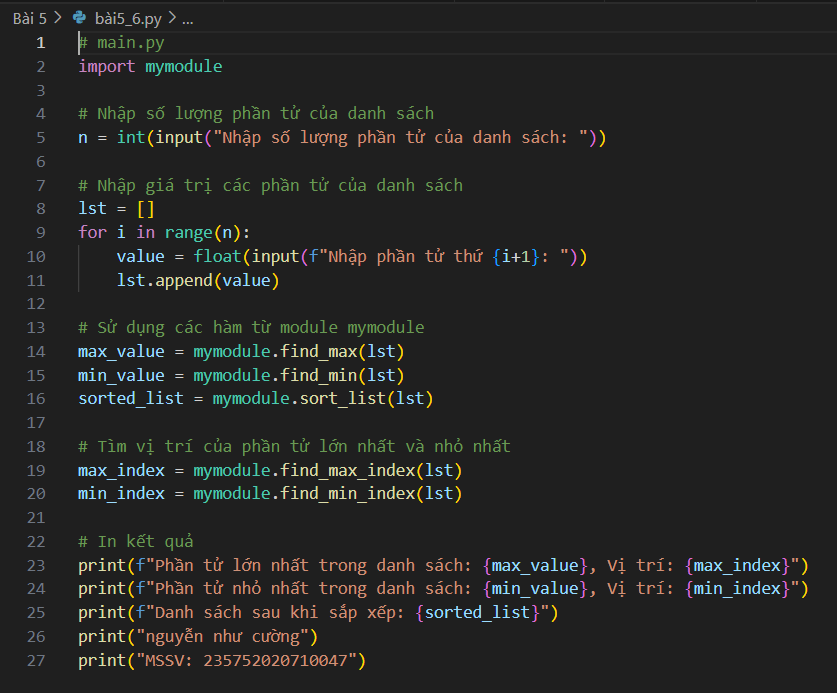
Câu 5.5. Viết chương trình tìm phần tử lớn nhất và nhỏ nhất của một danh sách  
- Số lượng và giá trị của list được nhập từ bàn phím  
- Phương thức sắp xếp và tìm phần tử lớn nhất được viết thành module



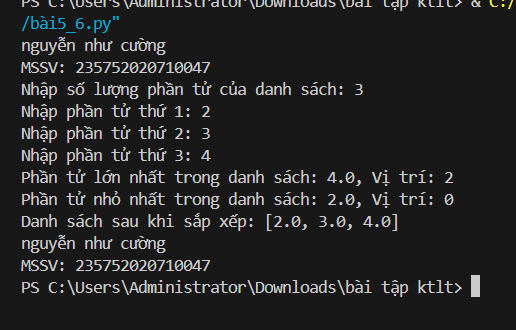
Kết quả:



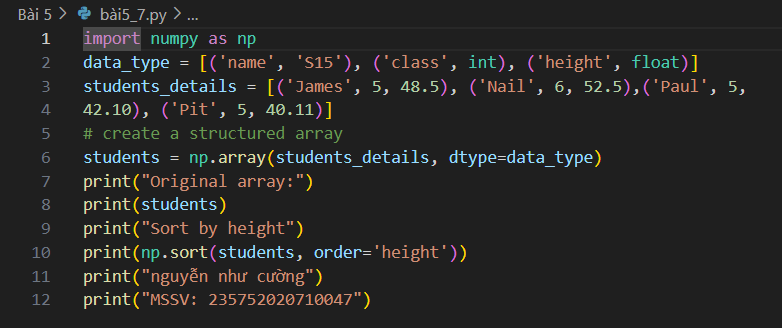
Câu 5.6. In ra vị trí phần tử lớn nhất và nhỏ nhất tìm được ở bài tập trên



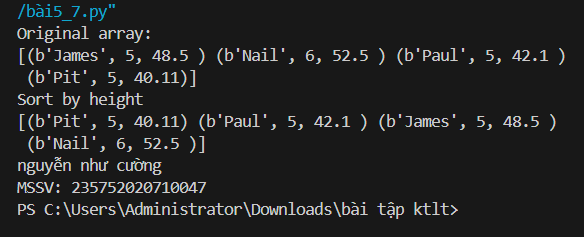
Kết quả:

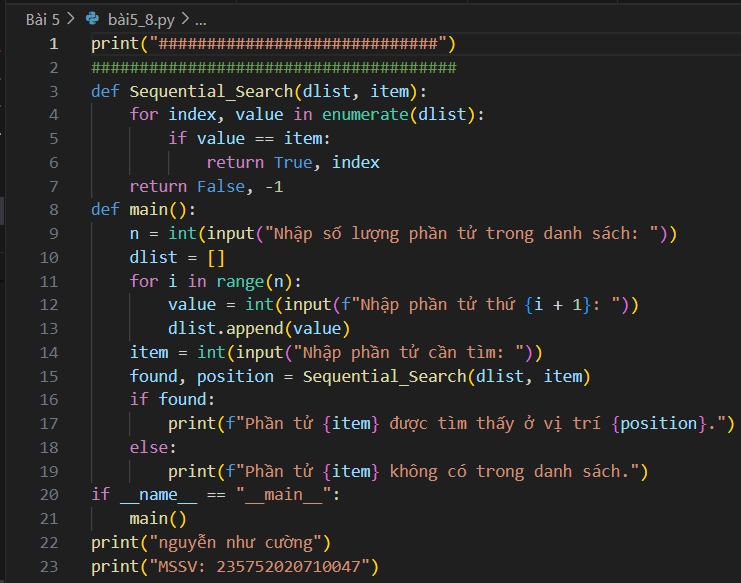


Câu 5.7. Viết chương trình sử dụng thư viện NumPy để tạo một mảng có cấu trúc từ tên sinh viên, chiều cao, lớp và các kiểu dữ liệu của họ. Bây giờ sắp xếp các mảng theo chiều cao.

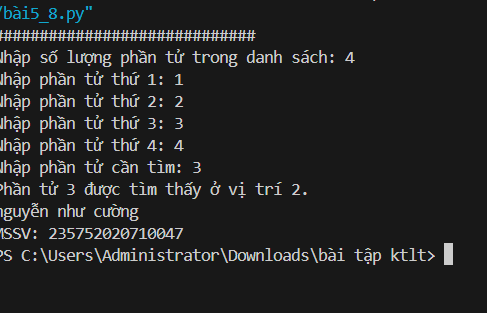


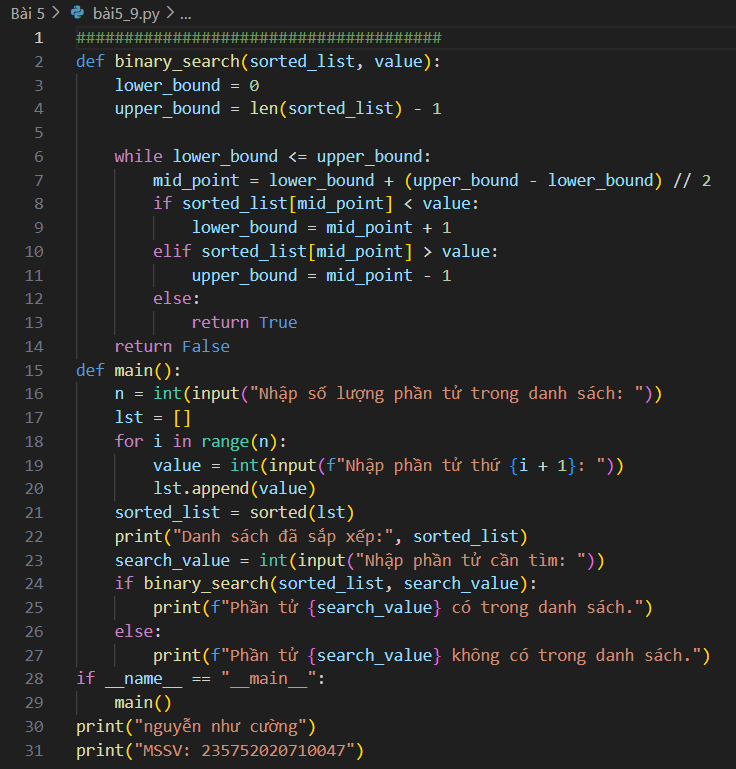
Kết quả:



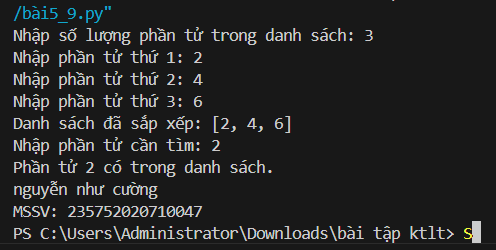
Câu 5.8. Xây dựng hàm “Sequential\_Search(dlist, item)” (giải thuật tìm kiếm tuyến tính) dưới dạng module. Viết chương trình nhập một dlist n phần tử từ bàn phím và tìm kiếm phần tử item bất kỳ. Sequential\_Search([11,23,58,31,56,77,43,12,65,19],31) -> (True, 3)  


Kết quả:

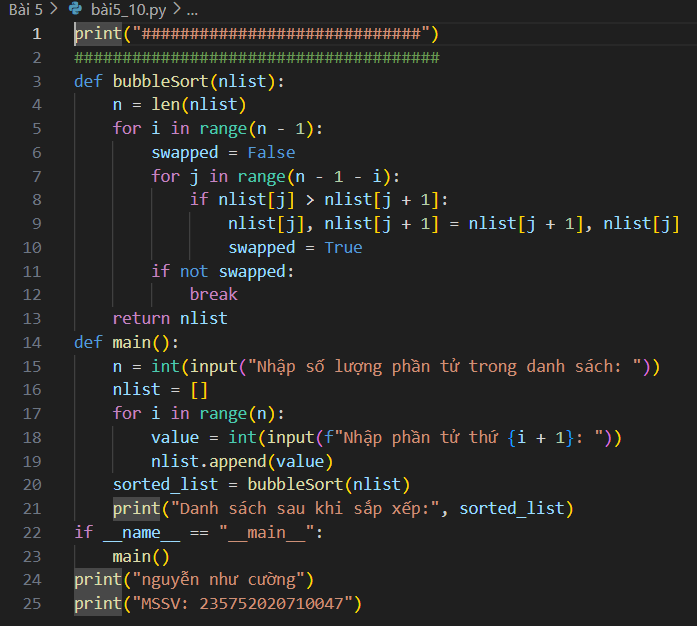


Câu 5.9. Xây dựng hàm *“binary\_search(list, value)”* (giải thuật tìm kiếm nhị phân) dưới dạng module. Viết chương trình nhập một list n phần tử từ bàn phím và tìm kiếm phần tử *value* bất kỳ.  
binary\_search([1,2,3,5,8], 6) -> False  
binary\_search([1,2,3,5,8], 5) -> True  


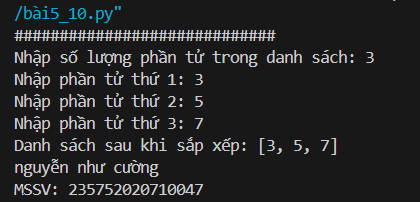
Kết quả:

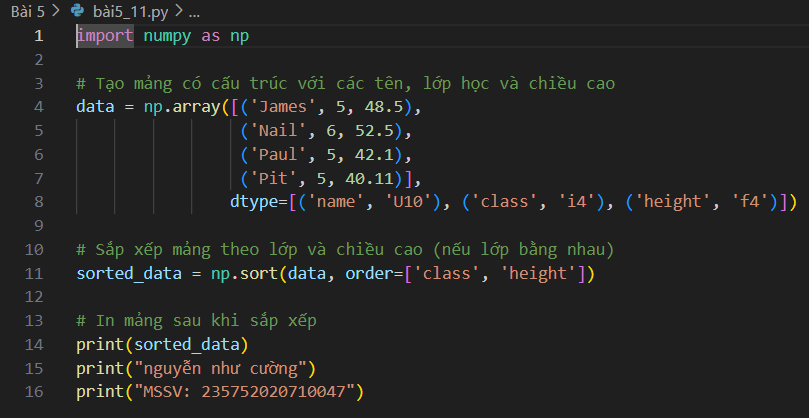


Câu 5.10 Xây dựng hàm *“bubbleSort (nlist)”* (giải thuật sắp xếp nổi bọt) dưới dạng module. Viết chương trình nhập một *nlist* n phần tử từ bàn phím và sắp xếp.  
Sample Data: [14,46,43,27,57,41,45,21,70]  
Expected Result: [14, 21, 27, 41, 43, 45, 46, 57, 70]

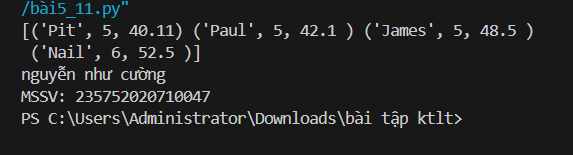


Kết quả:



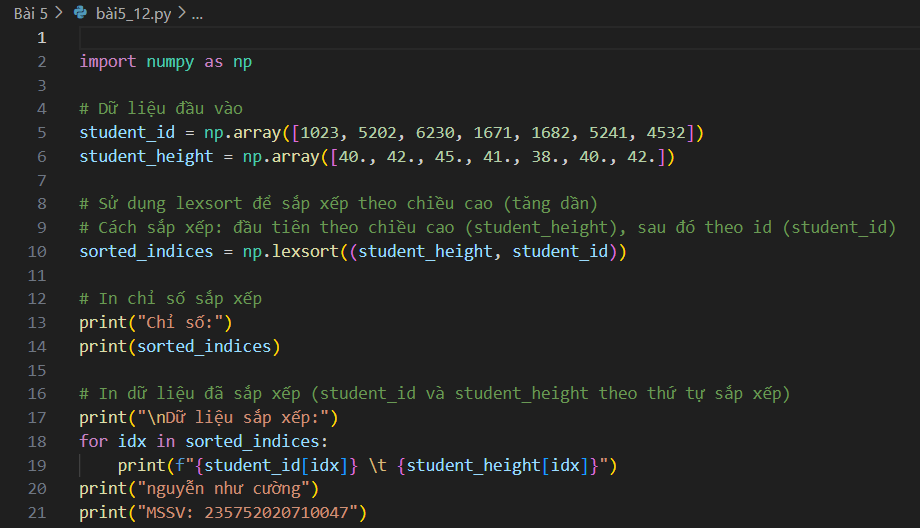
Câu 5.11. Viết chương trình sử dụng thư viện NumPy để tạo một mảng có cấu trúc từ tên sinh viên, chiều cao, lớp và các kiểu dữ liệu của họ. Bây giờ sắp xếp theo lớp, sau đó chiều cao nếu lớp bằng nhau.  
Dữ liệu đầu vào:  
[('James', 5, 48.5 ) ('Nail', 6, 52.5 ) ('Paul', 5, 42.1 ) ('Pit', 5, 40.11)]  
Kết quả sắp xếp:  
[('Pit', 5, 40.11) ('Paul', 5, 42.1 ) ('James', 5, 48.5 ) ('Nail', 6, 52.5 )]  


Kết quả:

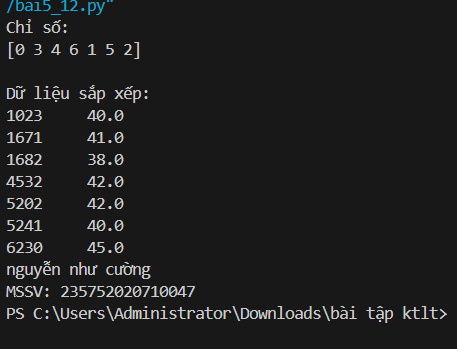


Câu 5.12. Viết chương trình sử dụng thư viện NumPy để sắp xếp id sinh viên với chiều cao tăng dần của sinh viên từ id sinh viên và chiều cao đã cho. In các chỉ số nguyên mô tả thứ tự sắp xếp theo nhiều cột và dữ liệu được sắp xếp (sử dụng hàm *lexsort()*)

Dữ liệu đầu vào:  
student\_id = [1023, 5202, 6230, 1671, 1682, 5241, 4532]  
student\_height = [40., 42., 45., 41., 38., 40., 42.0]



Kết quả:



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

**6.1. Mục đích**

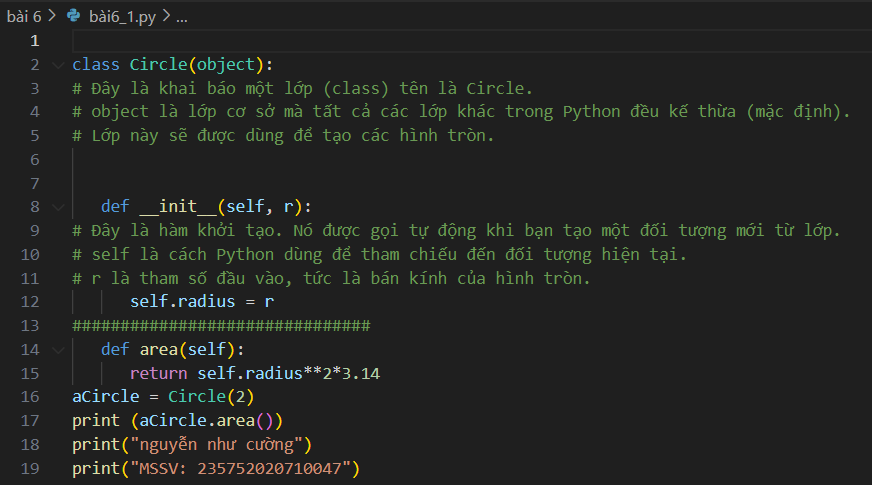
Giúp sinh viên nắm bắt về lập trình hướng đối tượng trong lập trình python.

**6.2. Cơ sở lý thuyết**

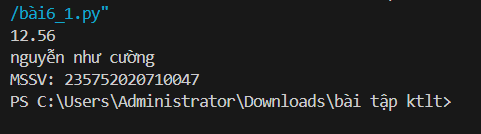
Xem các quy tắc xây dựng class, các phương thức và thuộc tính của đối tượng

**6.3. Các bước tiến hành**

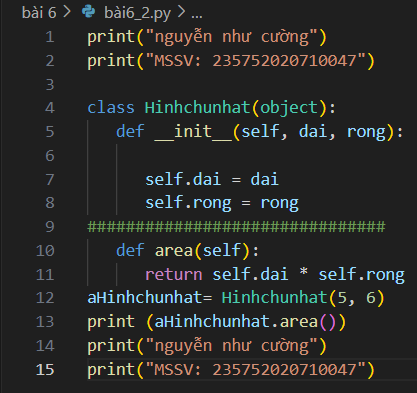
Câu 6.1. Định nghĩa một class có tên là Circle có thể được xây dựng từ bán kính. Circle có một method có thể tính diện tích.



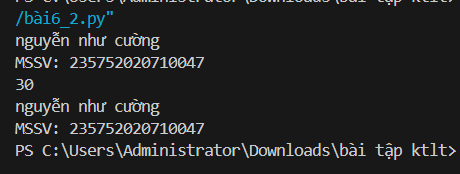
Kết quả:



Câu 6.2. Định nghĩa class có tên là *Hinhchunhat* được xây dựng bằng chiều dài và chiều rộng. Class Hinhchunhat có method để tính diện tích.



Kết quả:

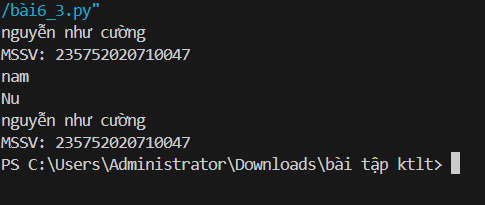


Câu 6.3. Định nghĩa class *Nguoi* và 2 class con của nó: *Nam, Nu*. Tất cả các

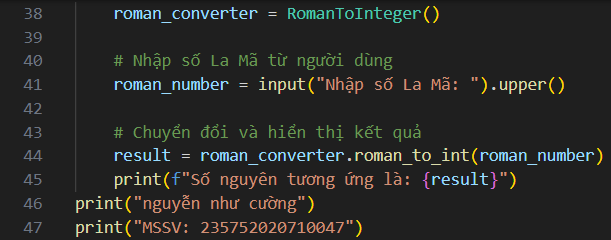
class có method "*getGender*" có thể in "*Nam*" cho class *Nam* và "*Nữ*" cho class *Nu*



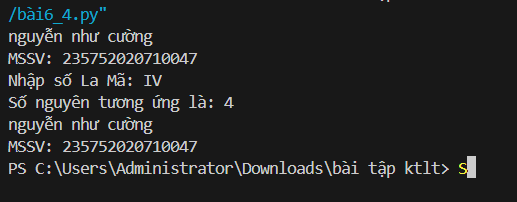
Kết quả:



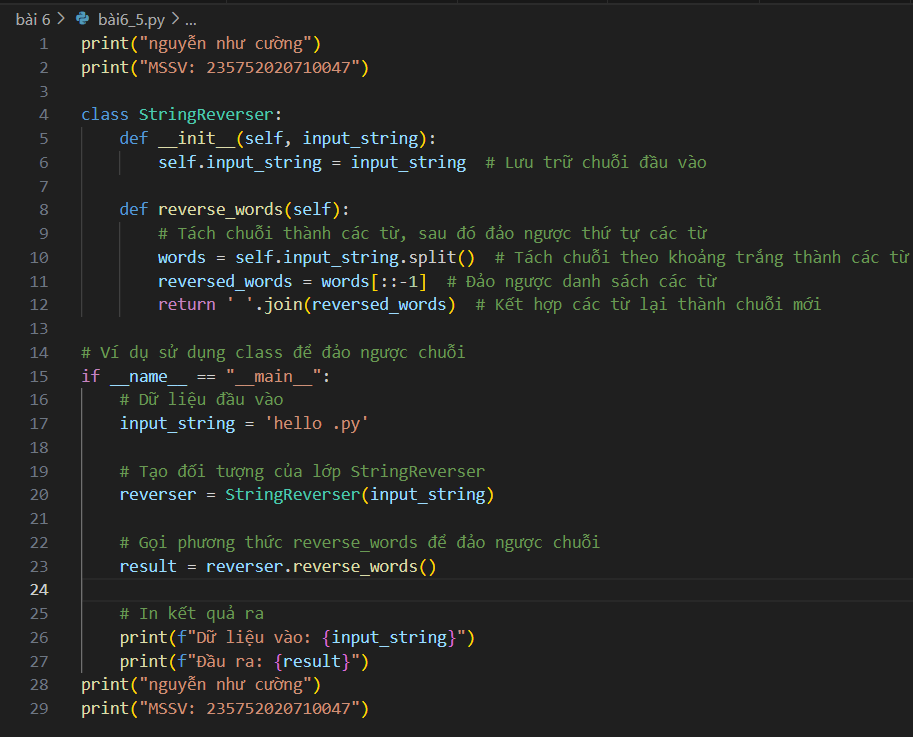
Câu 6.4. Viết chương trình Python dưới dạng class để chuyển đổi một số La Mã thành một số nguyên.

Kết quả:



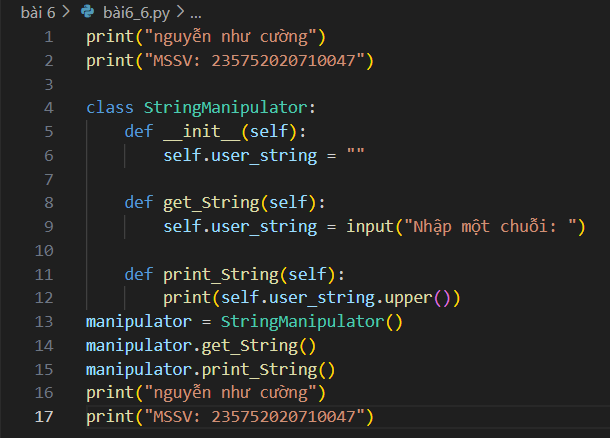
Câu 6.5. Viết chương trình Python dưới dạng class để đảo ngược chuỗi từ từng chữ.  
Dữ liệu vào : 'hello .py'  
Đầu ra : '.py hello



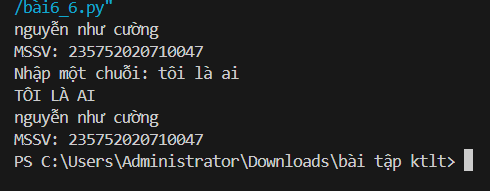
Kết quả:



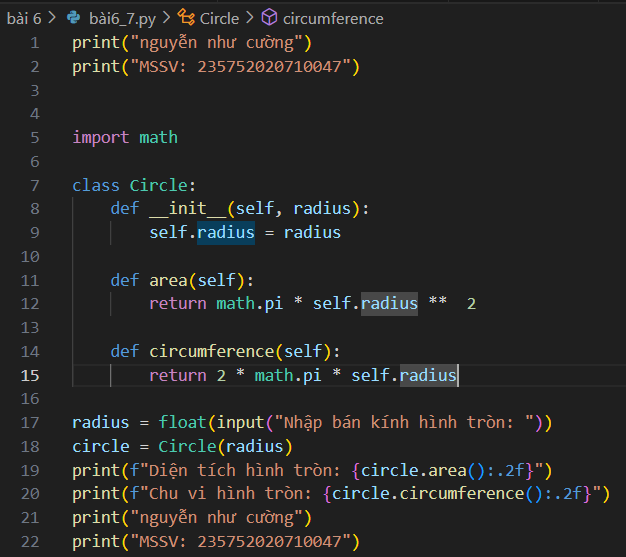
Câu 6.6. Viết một class Python có hai phương thức get\_String và print\_String. get\_String chấp nhận một chuỗi từ người dùng và print\_String in chuỗi đó bằng chữ in hoa.



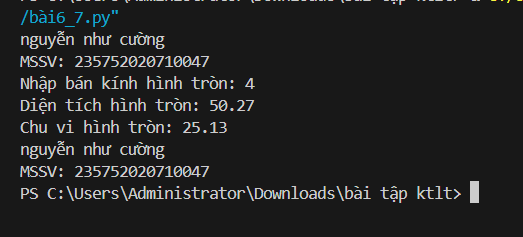
Kết quả:



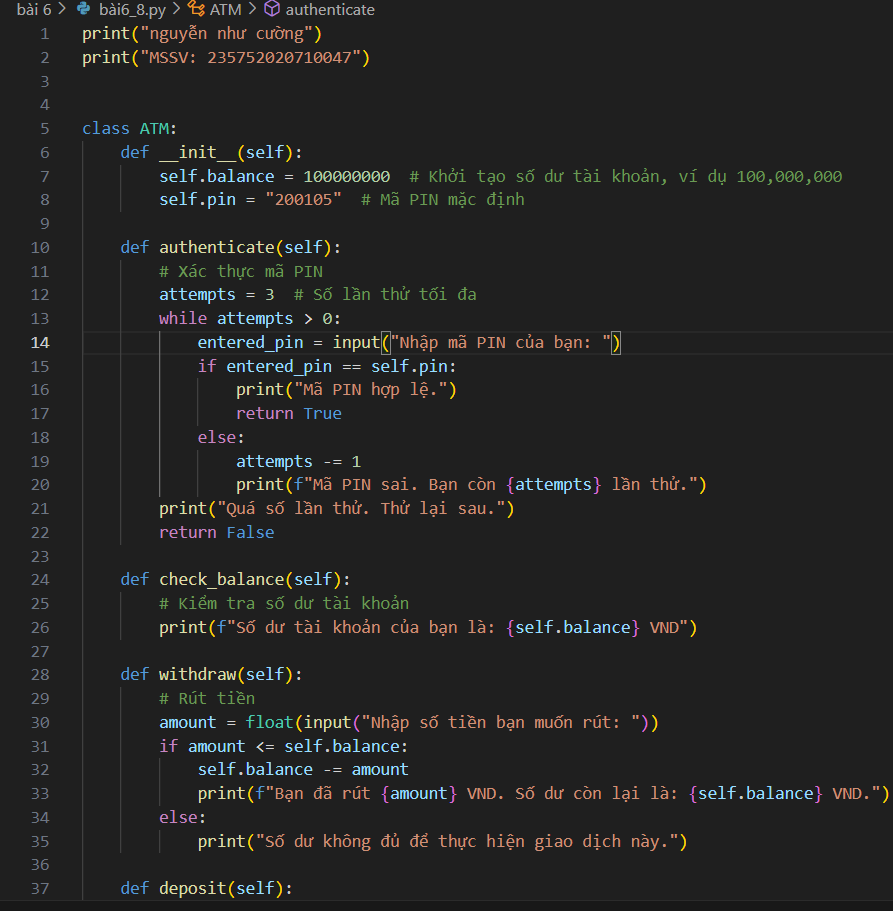
Câu 6.7. Viết một class Python có tên Circle được xây dựng theo bán kính và hai phương thức sẽ tính diện tích và chu vi của hình tròn.

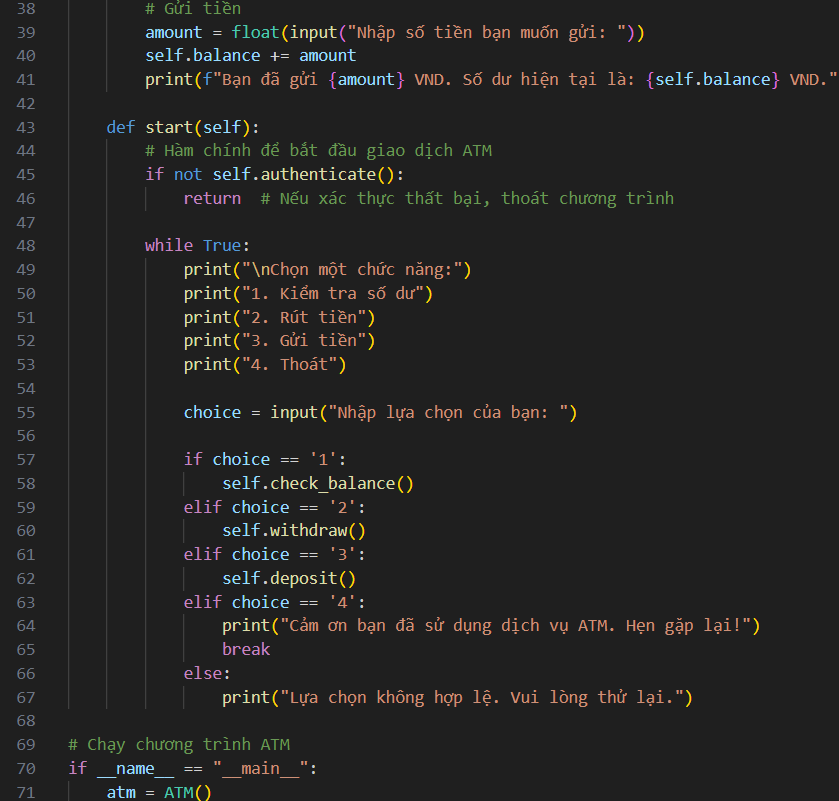


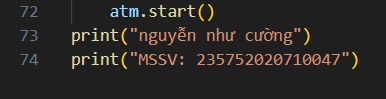
Kết quả:



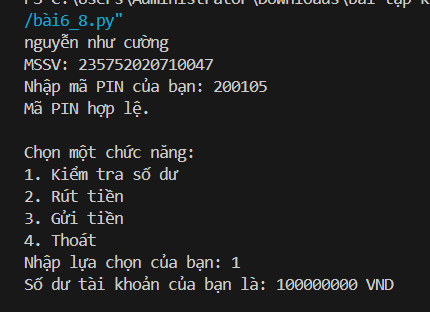
Câu 6.8. Chương trình ATM đơn giản







Kết quả:



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

**7.1. Mục đích**

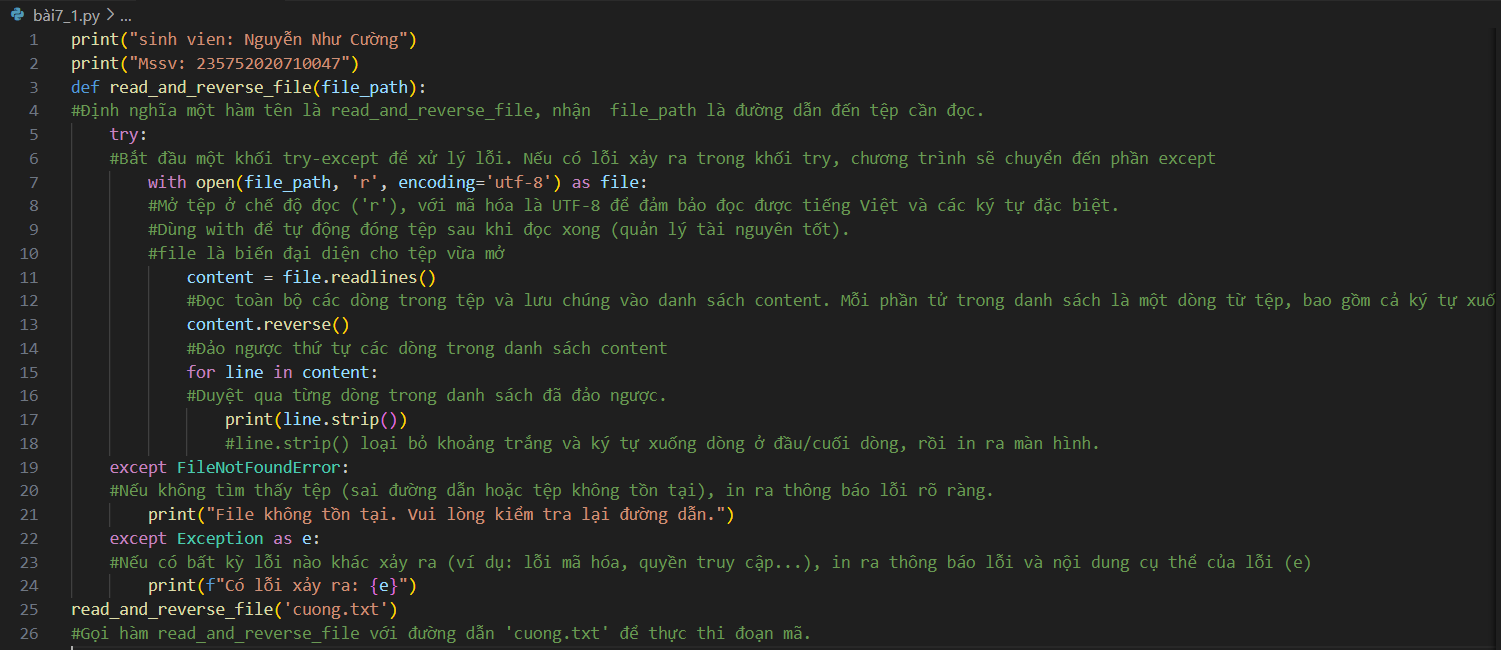
Giúp sinh viên có thể thao tác với các file văn bản sử dụng python.

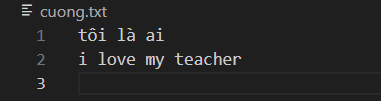
**7.2.** **Cơ sở lý thuyết**

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

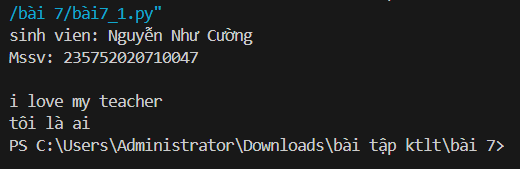
**7.3. các bước tiến hành**

Câu 7.1. Chương trình đọc file và in đảo ngược kết quả

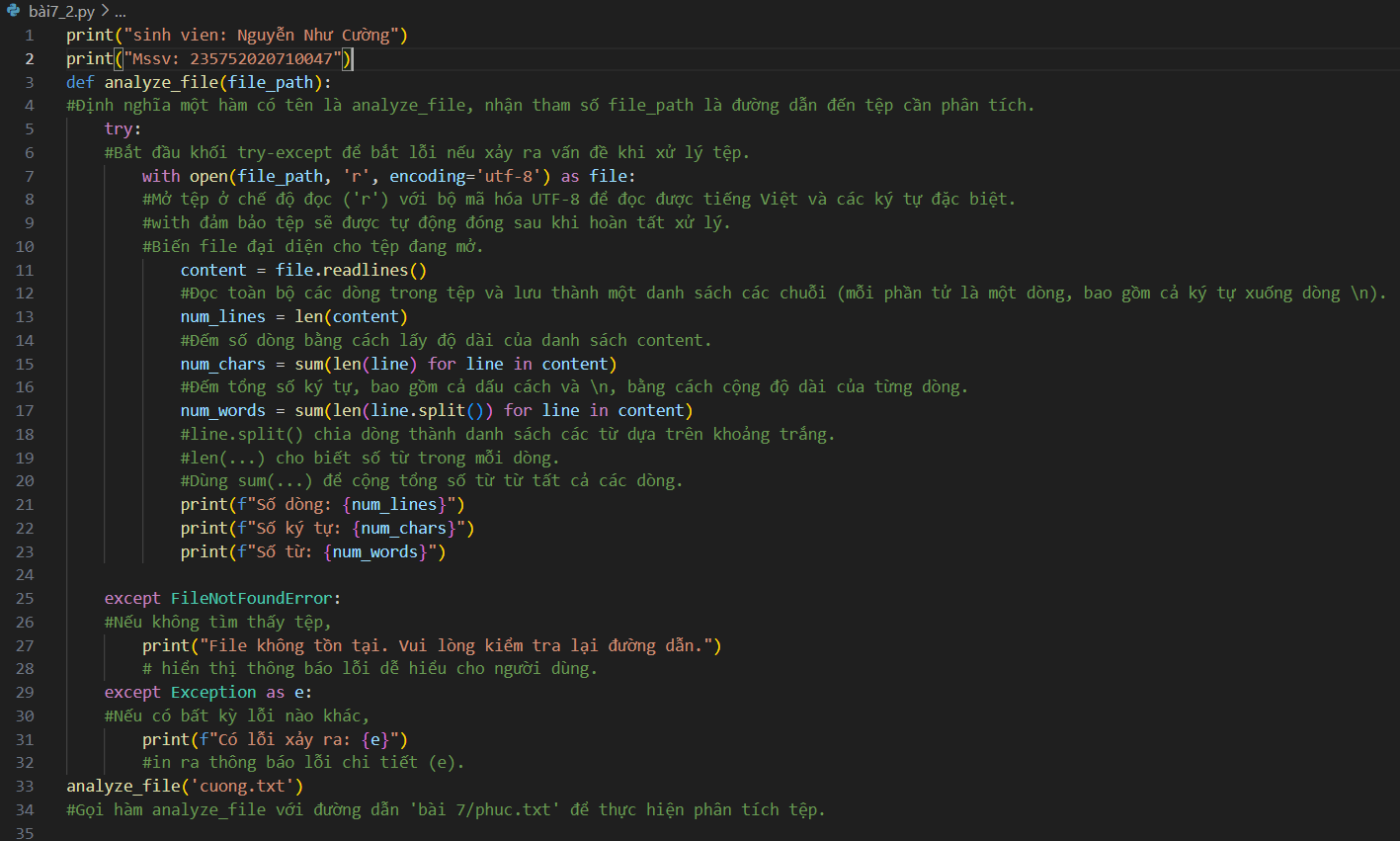




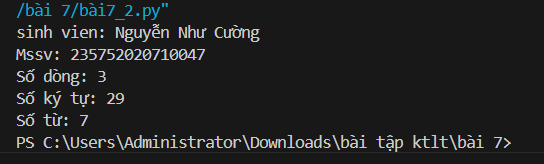
Kết quả:



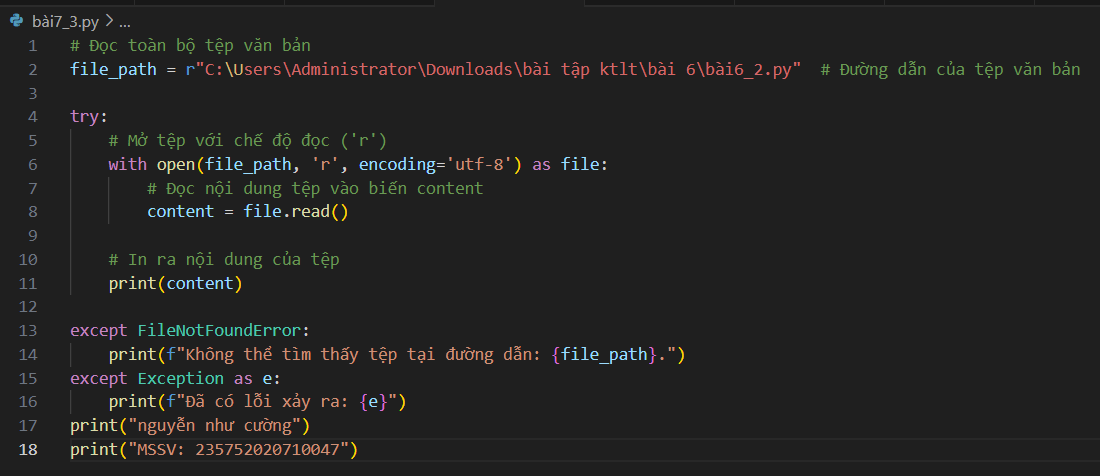
Câu 7.2. Chương trình đọc một file, tính số ký tự, số từ và số dòng của file



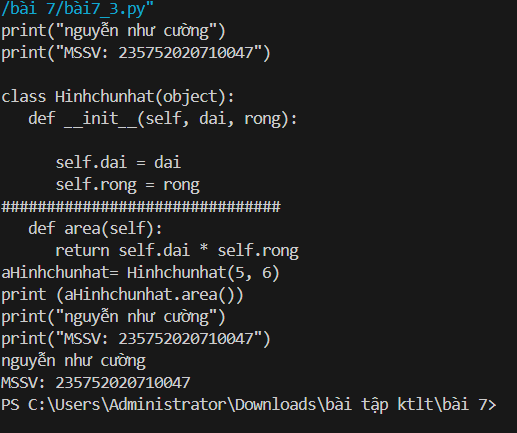
Kết quả:



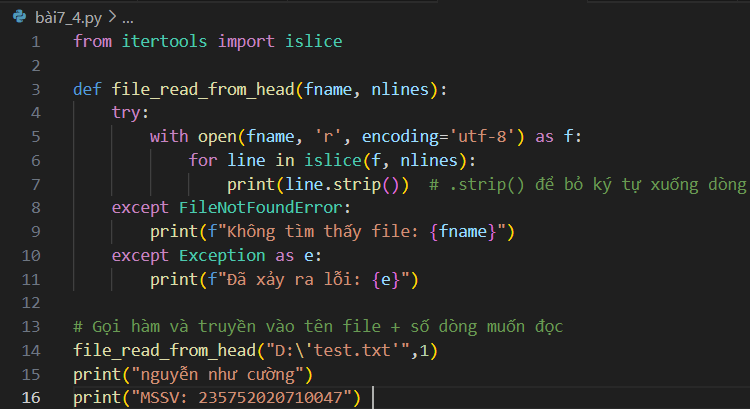
Câu 7.3. Viết chương trình Python để đọc toàn bộ tệp văn bản



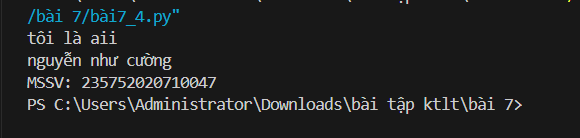
Kết quả:



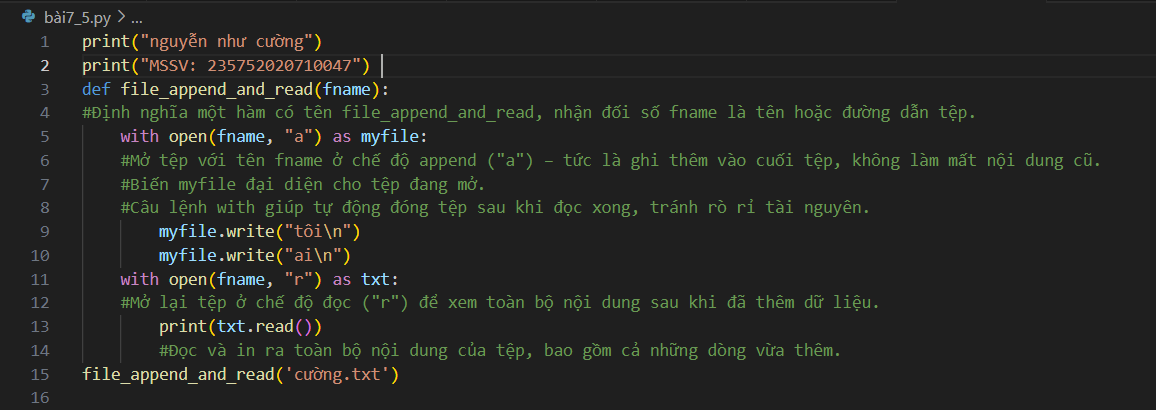
Câu 7.4. Chương trình Python để đọc n dòng đầu tiên của tệp



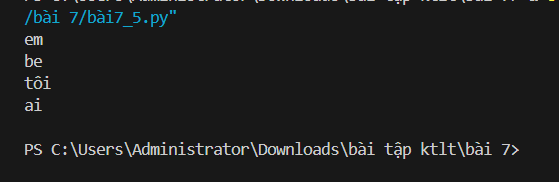
Kết quả:



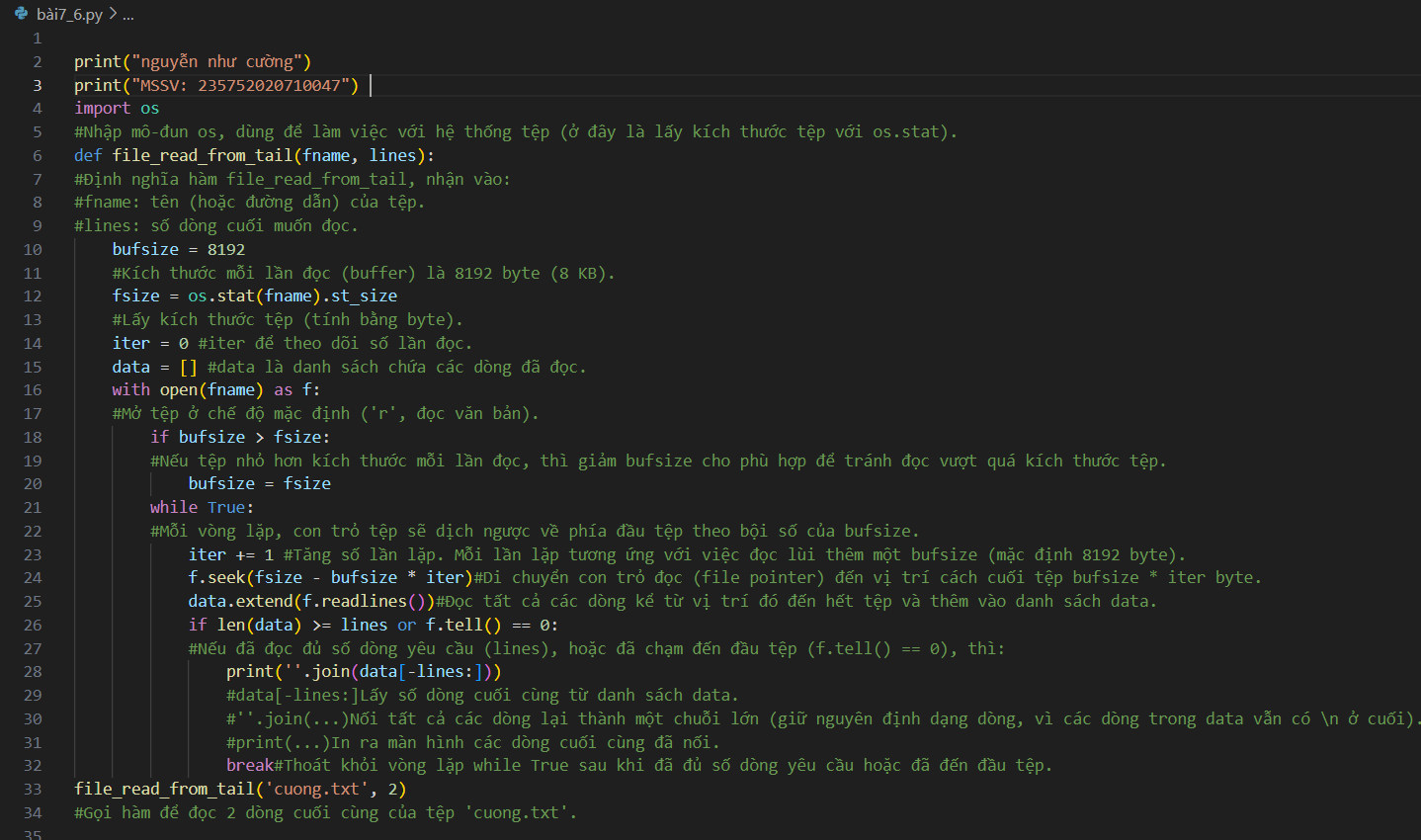
Câu 7.5. Chương trình Python để nối văn bản vào tệp và hiển thị văn bản



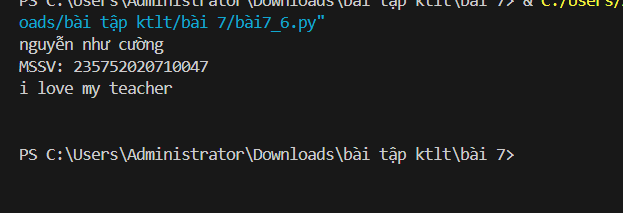
Kết quả:



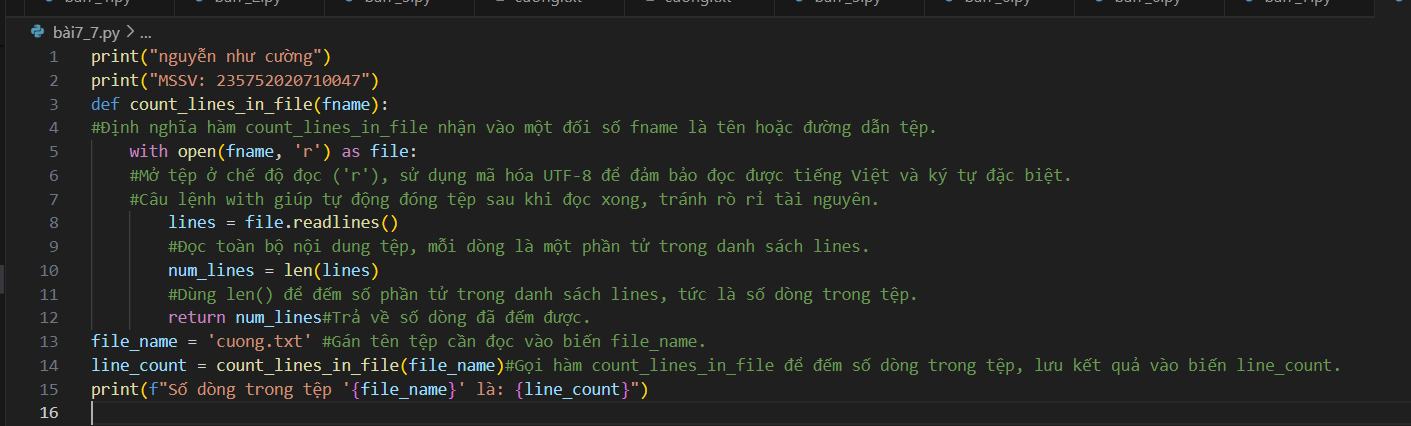
Câu 7.6. Chương trình Python để đọc n dòng cuối cùng của tệp



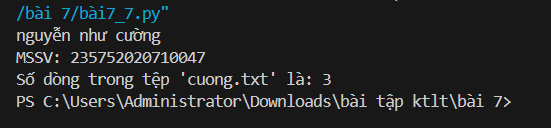
Kết quả:



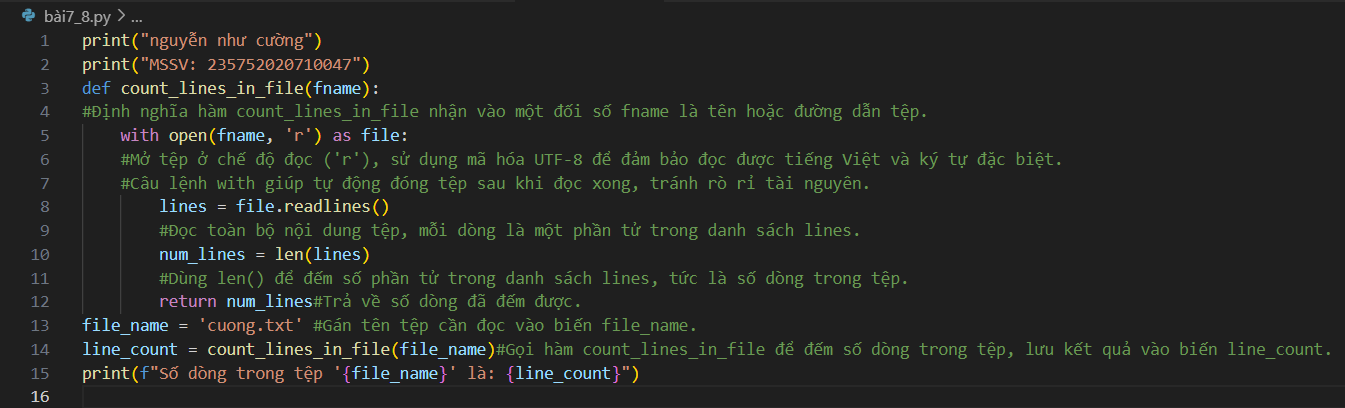
Câu 7.7. Viết chương trình Python để đếm số dòng trong tệp văn bản



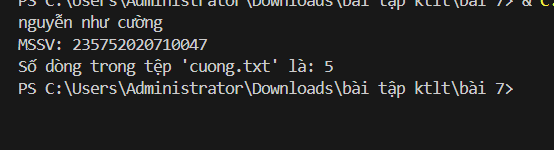
Kết quả:



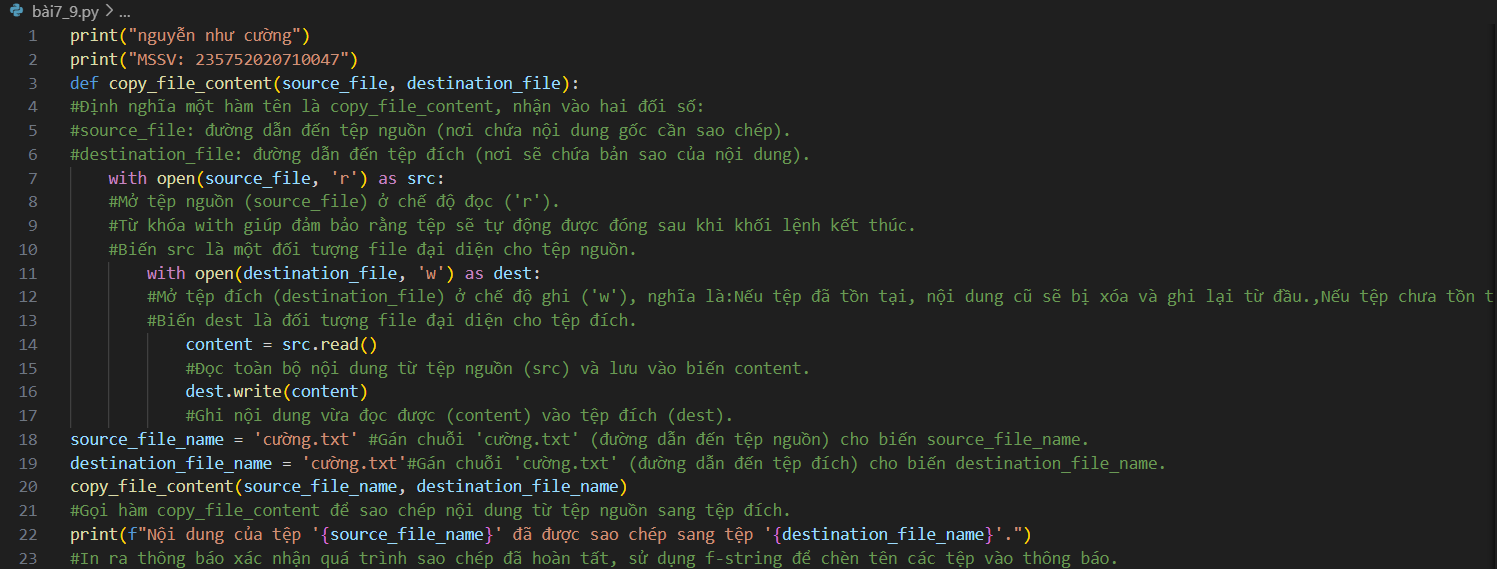
Câu 7.8. Viết chương trình Python để viết nội dung danh sách vào tệp.



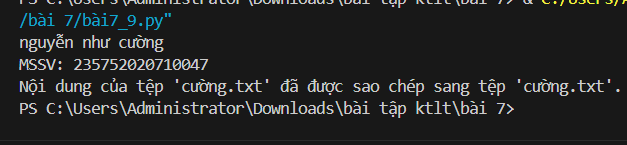
Kết quả:



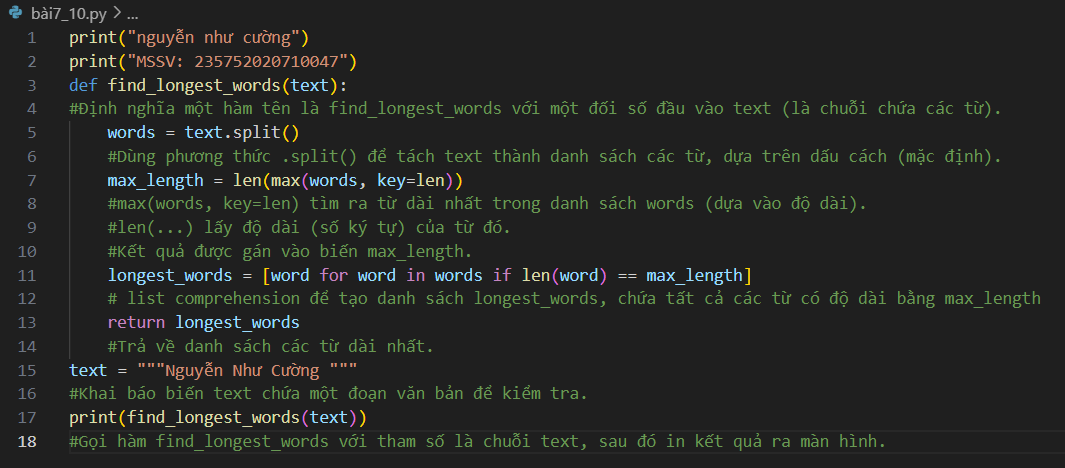
Câu 7.9. Viết chương trình Python để sao chép nội dung của tệp này sang tệp khác.



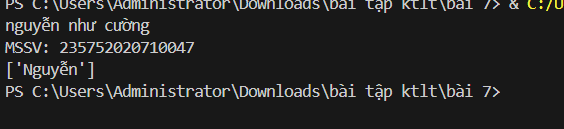
Kết quả:



Câu 7.10. Viết chương trình python để tìm những từ dài nhất trong văn bản



Kết quả:



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

**8.1. Mục đích**

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

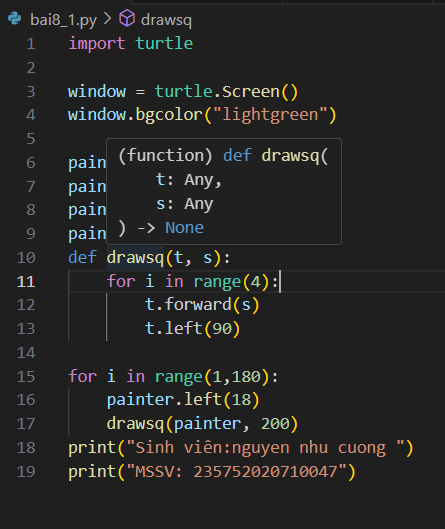
**8.2. Cơ sở lý thuyết**

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

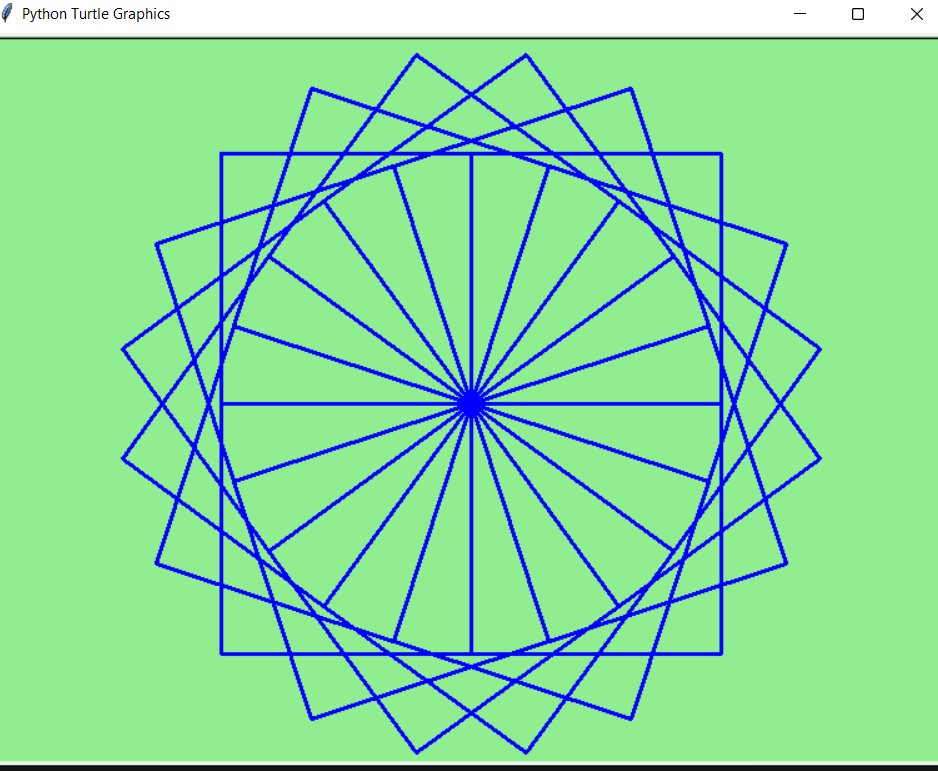
Pỵthon

**8.3. Các bước thực hành**

**Câu 8.1. Viết chương trình đồ họa sử dụng thư viện turtle, kiểm tra kết quả và giải thích chương trình.**

****

*Kết quả:*

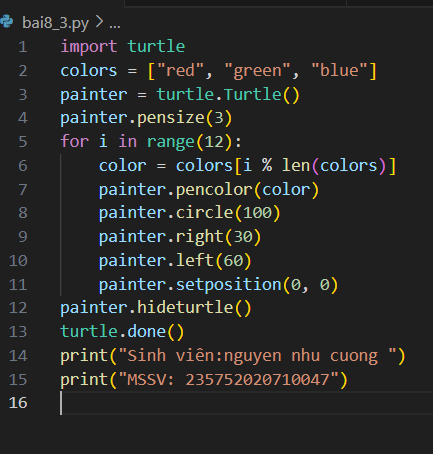


Câu 8.2.

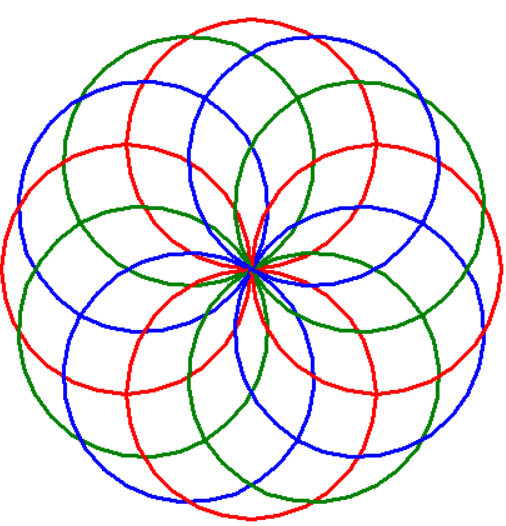


Kết quả:

Câu 8.3. Dựa trên các kết quả đạt được từ các chương trình trên hãy viết chương trình hiển thị hình ảnh đồ họa sau

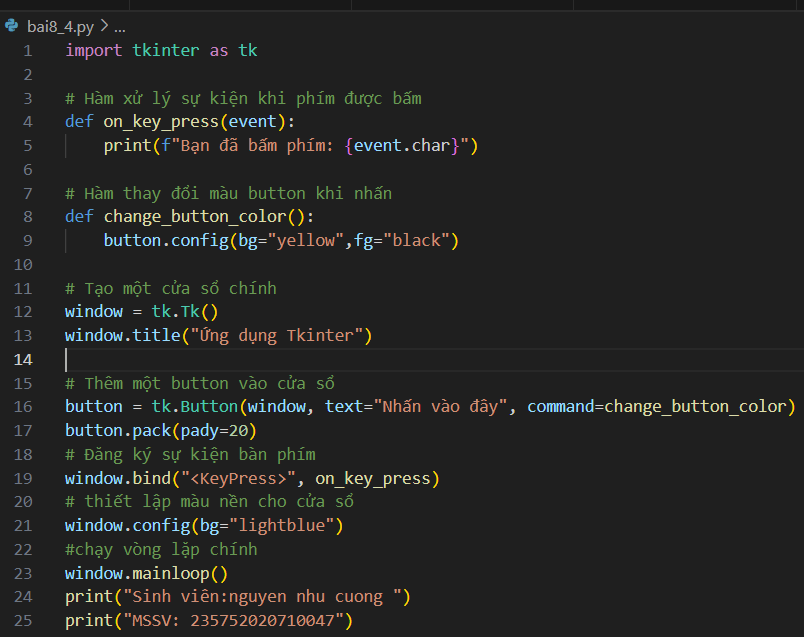


Kết quả:

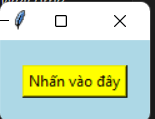


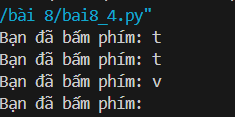
Câu 8.4. Viết chương sử dụng thư viện đồ họa tkinter thực hiện:  
a) Xây dựng cửa sổ đồ họa window form  
b) Thêm một widget (button) vào window form  
c) Xây dựng phương thức xử lý sự kiện phím bấm

d) Thay đổi màu nền và màu chữ của button sử dụng thuộc tính “bg” và “fg’’



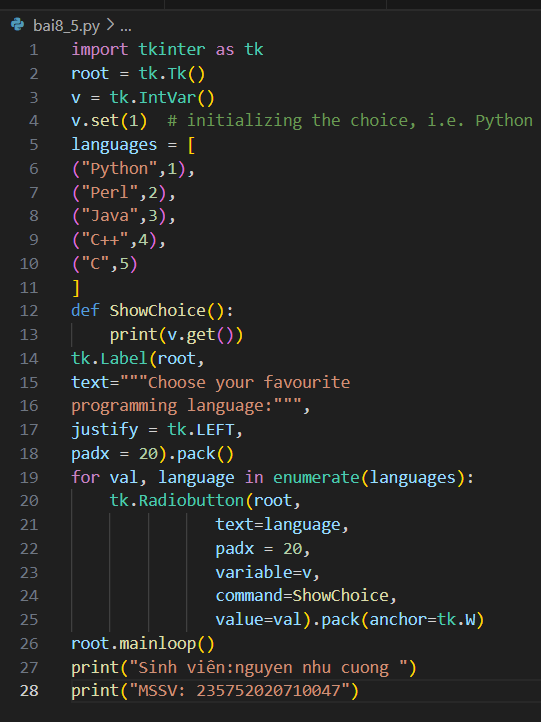
Kết quả:



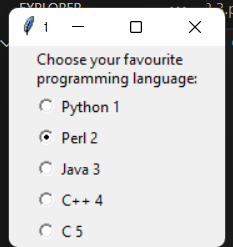


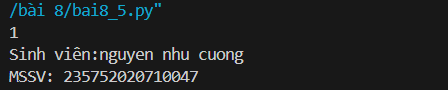
Câu 8.5. Sử dụng thư viện tkinter thực hiện:  
a) Xây dựng các radio button cho phép thực hiện các lựa chọn khác nhau

b) Thay thế các radio button thành các indicator như hình

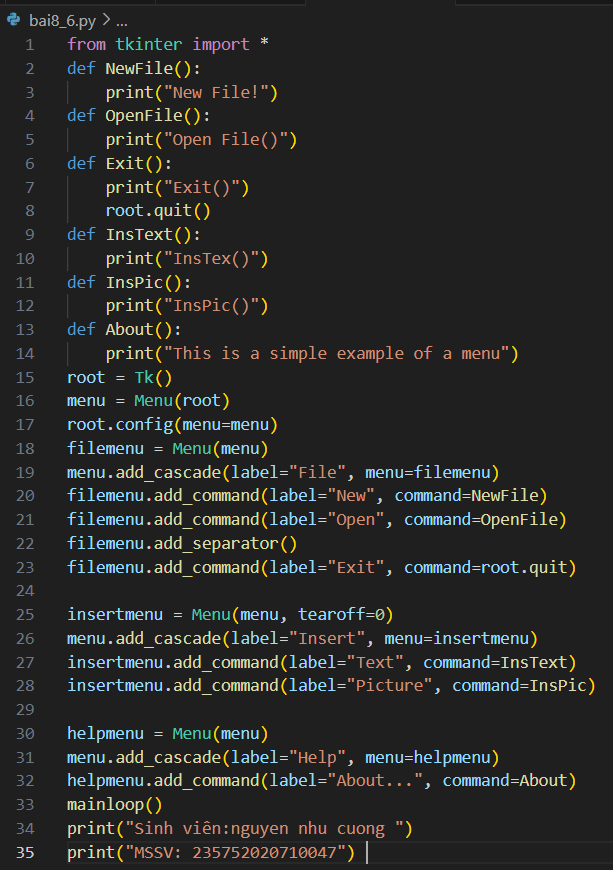


Kết quả:

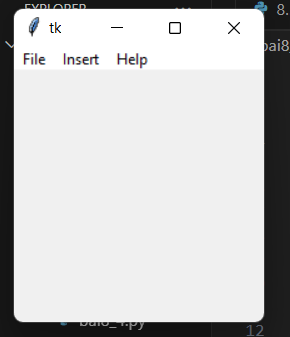




Câu 8.6. Viết chương trình thực hiện tạo menu

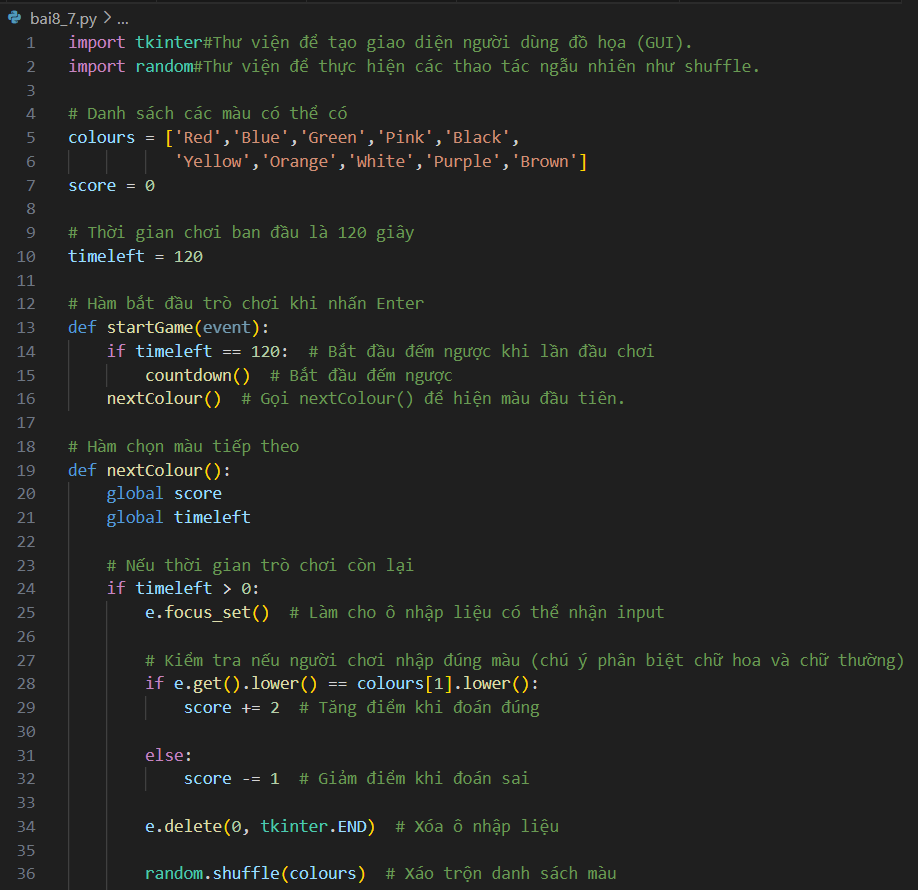


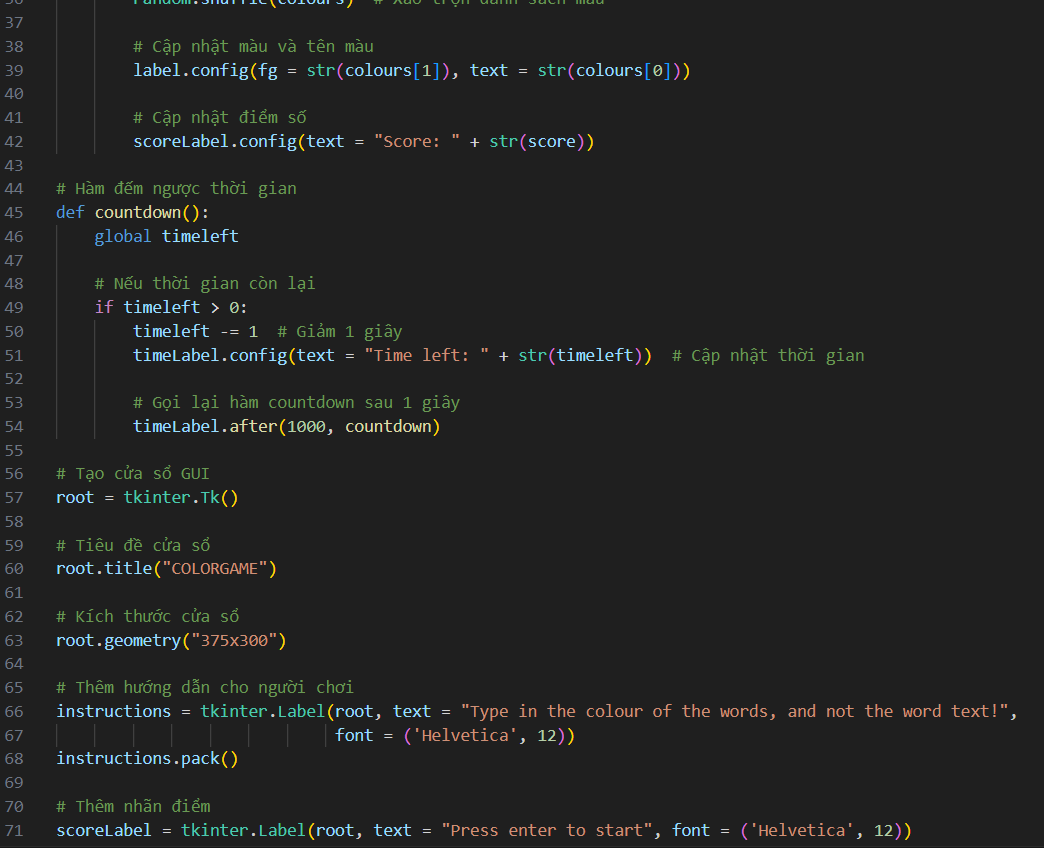
Kết quả:

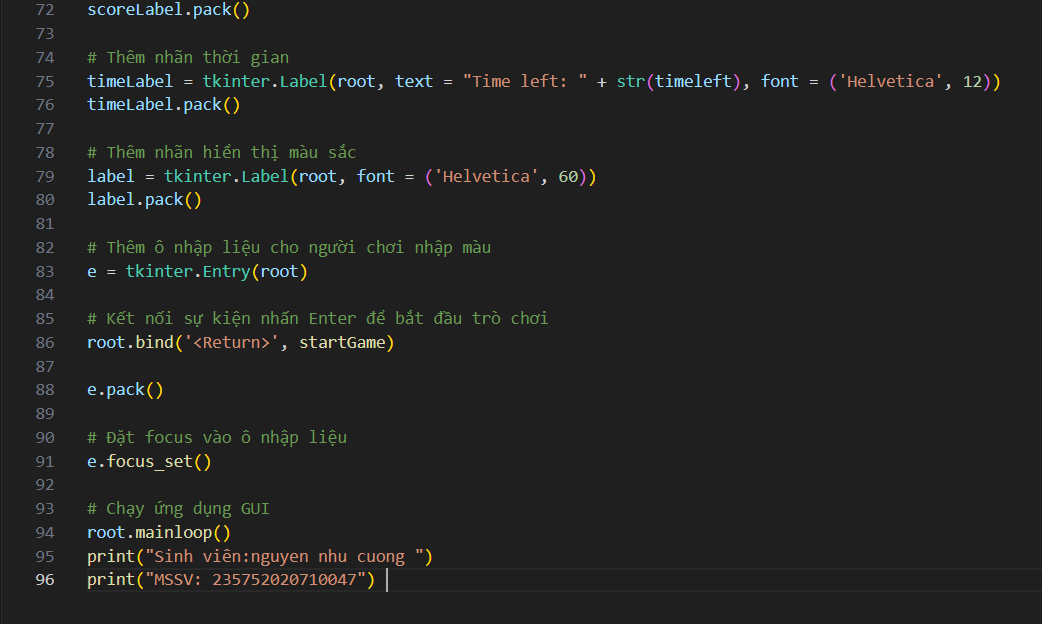


Câu 8.7. Sử dụng thư viện đồ họa tkinter thực hiện các bước sau  
Bước 1: Viết chương trình game học các màu tiếng anh

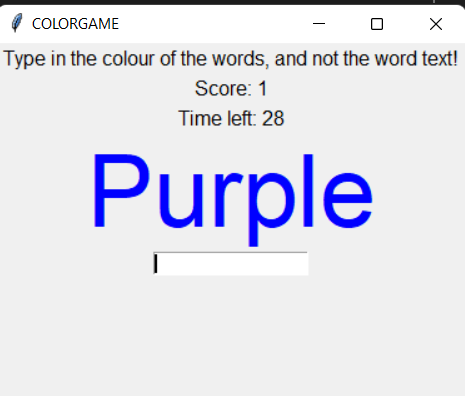
Bước 2: Thay đổi thời gian chơi từ 30s thành 120s  
Bước 3: Thay đổi số điểm cộng cho mỗi lần đoán đúng là 2, mỗi lần đoán sai là -1.



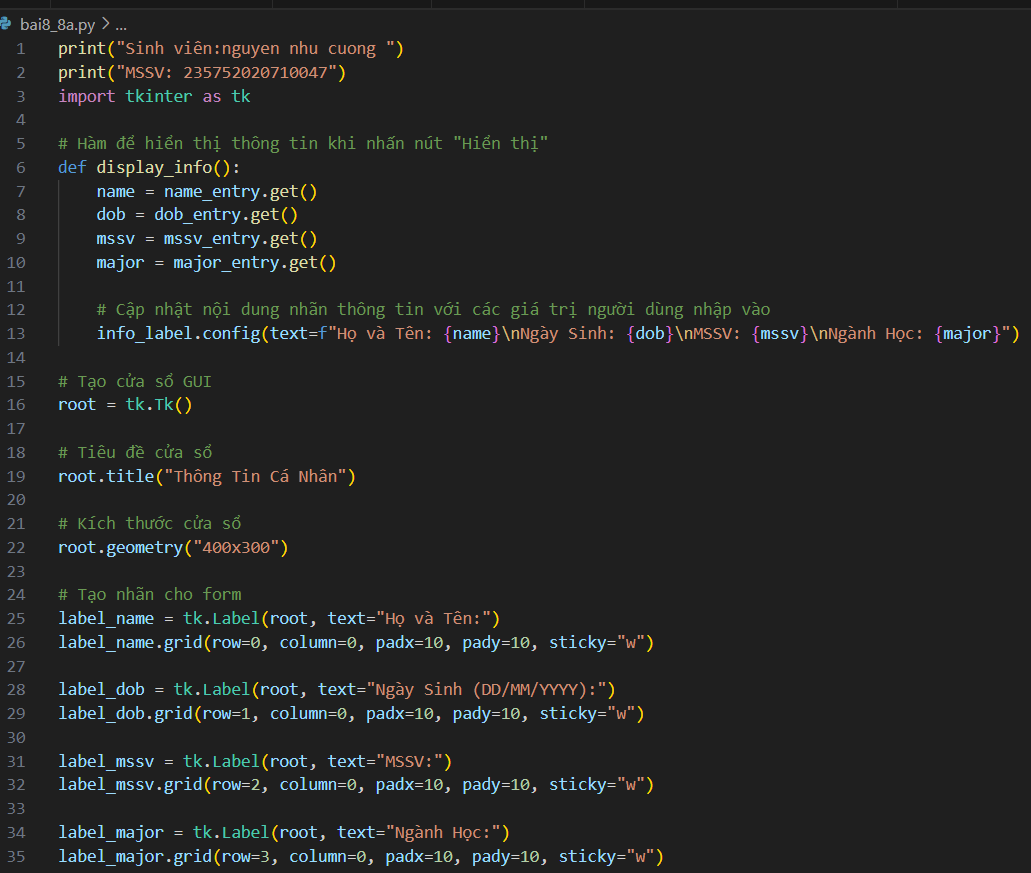


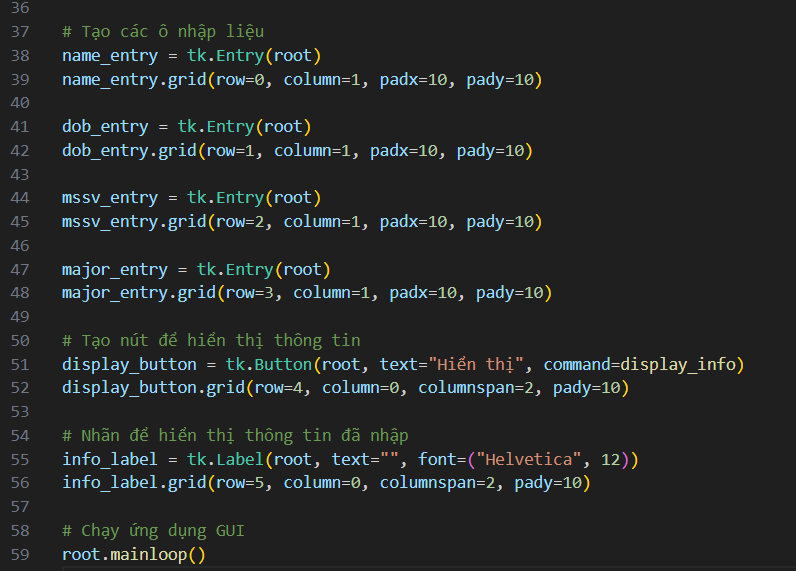


Kết quả

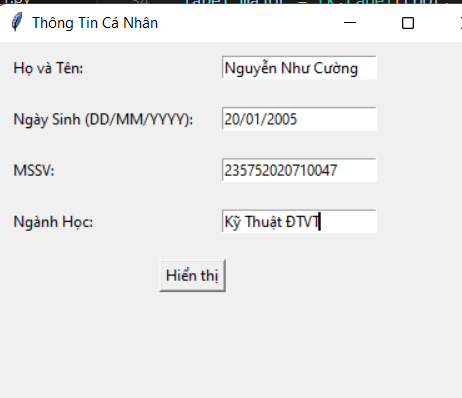


Câu 8.8. Viết chương trình graphic sử dụng thư viện Tkinter thực hiện:  
a) Xây dựng form hiển thị thôn tin cá nhân (họ tên, ngày tháng năm sinh, MSSV,  
ngành học)



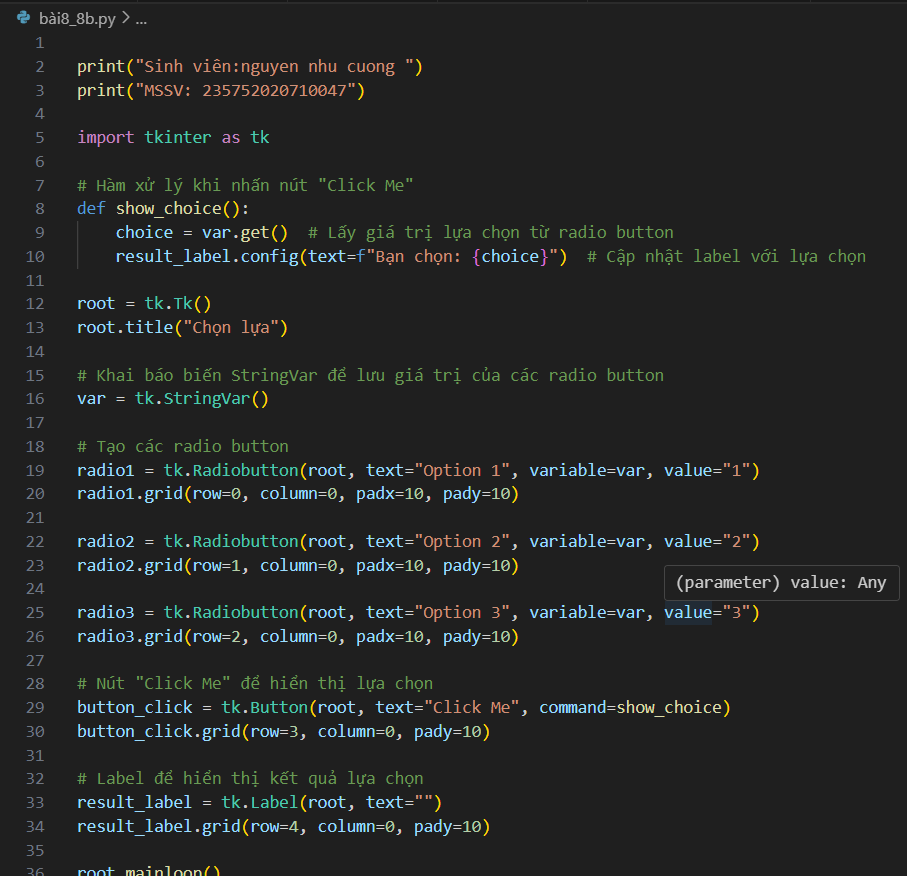


Kết quả:



b) Xây dựng form có nội dung như hình ở dưới, khi bấm vào nút “Click Me”  
thông tin nút radio button đang lựa chọn sẽ được chỉ ra (tương ứng với các số 1,  
2, 3)

Kết quả:



Kết quả

