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



BÁO CÁO THỰC HÀNH

**KỸ THUẬT LẬP TRÌNH**

MÃ HỌC PHẦN: ELE20004

SVTH: LÊ QUANG TRUNG

MSSV: 235752021610012

GVHD: MAI THẾ ANH

**NGHỆ AN - 2024**

**MỤC LỤC**

[Bài 1. Thực hiện các thuật toán bằng phần mềm Flowgorithm 2](#_Toc120959851)

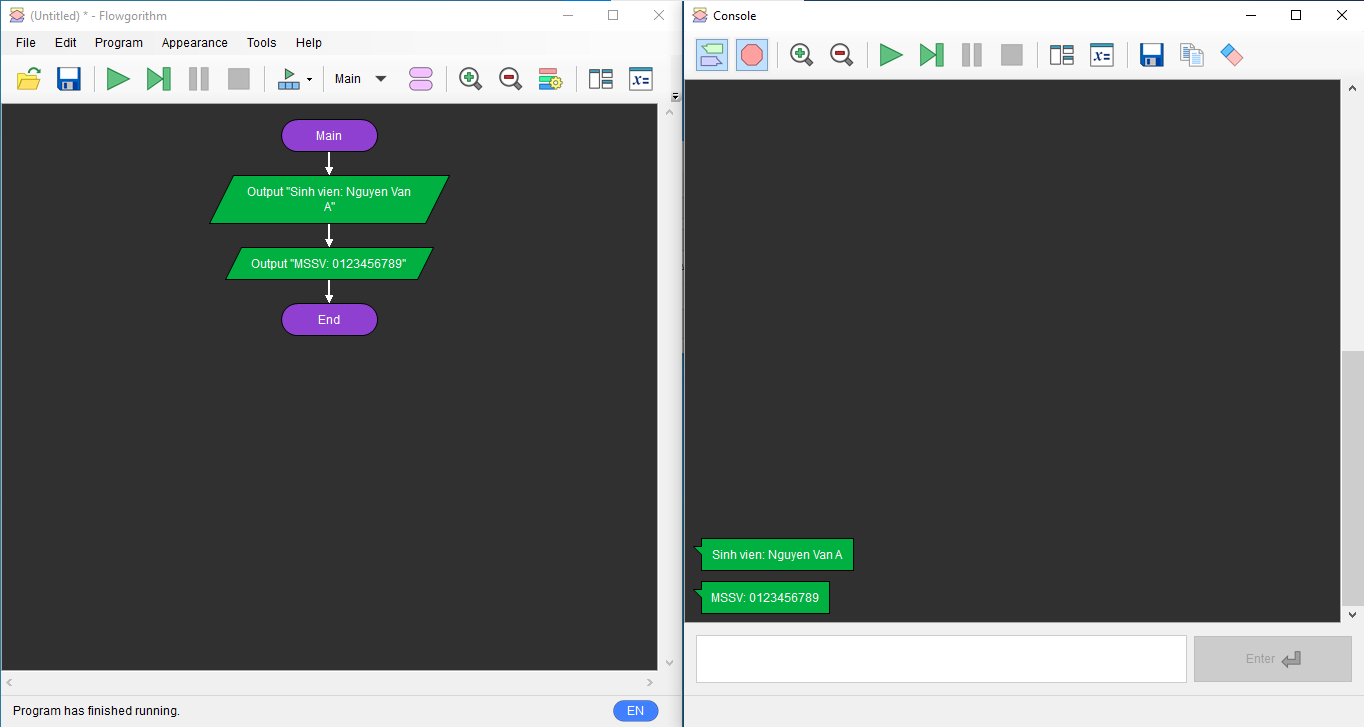
[Bài 2. Các cú pháp, kiểu dữ liệu, lệnh điều khiển trong lập trình Python 3](#_Toc120959852)

# **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)*

**

* 1. **Câu hỏi kiểm tra**

Trả lời ngắn gọn các câu hỏi theo yêu cầu.

* 1. **Tài liệu tham khảo**

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

* 1. **Mục đích**

........................................................... **Các bước**...........

* 1. **thực hiện và kết quả**

**1. Viết đoạn chương trình sau và sửa lỗi**

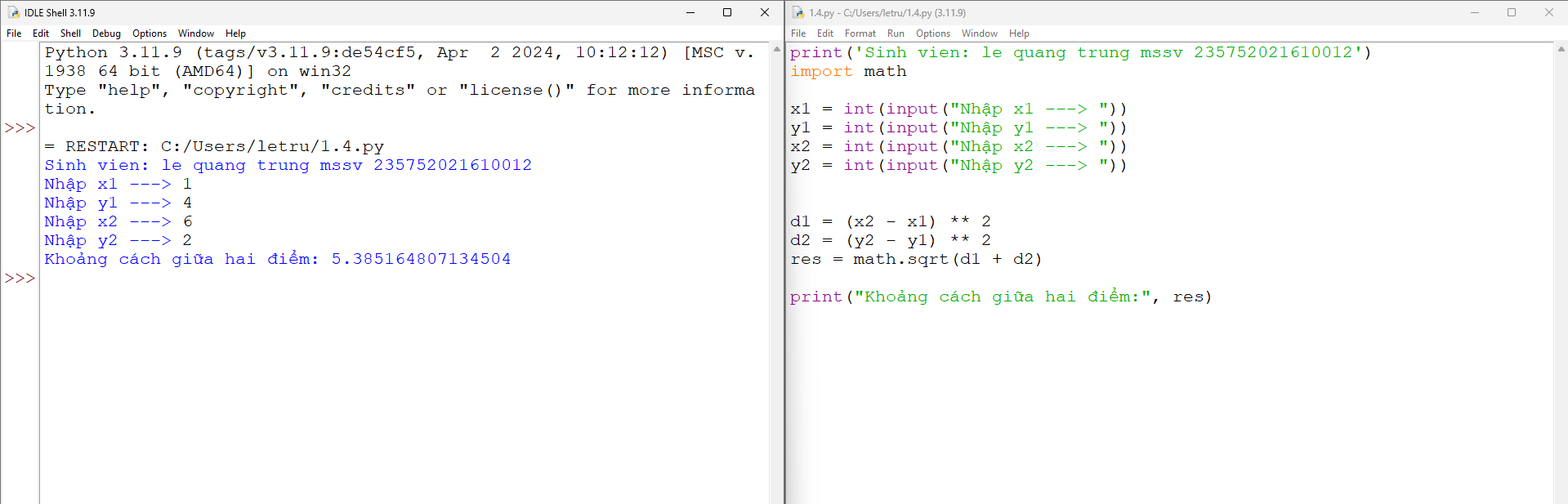
A screenshot of a computer

Description automatically generated

2.

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

1. Viết chương hai điểm và tính khoảng cách



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

A screenshot of a computer

Description automatically generated

1. 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)

A screenshot of a computer

Description automatically generated

1. 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

A screenshot of a computer

Description automatically generated

1. 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

A screenshot of a computer

Description automatically generated

1. 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. Ví dụ: Giả sử số n là 8 thì đầu ra sẽ là: {1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7:

49, 8: 64}

A screenshot of a computer

Description automatically generated

1. Viết chương trình in ra màn hình dãy số Fibonacci nhỏ hơn 4.000.000, tìm tổng

các số chẵn trong dãy đã in

A screenshot of a computer screen

Description automatically generated

1. 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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer

Description automatically generated

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

A screenshot of a computer screen

Description automatically generated

1. .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.

A screenshot of a computer code

Description automatically generated

1. Viết chương trình giải phương trình bậc 2: ax2

+bx+c=0, với các hệ số a, b, c nhập từ bàn phím

A screenshot of a computer screen

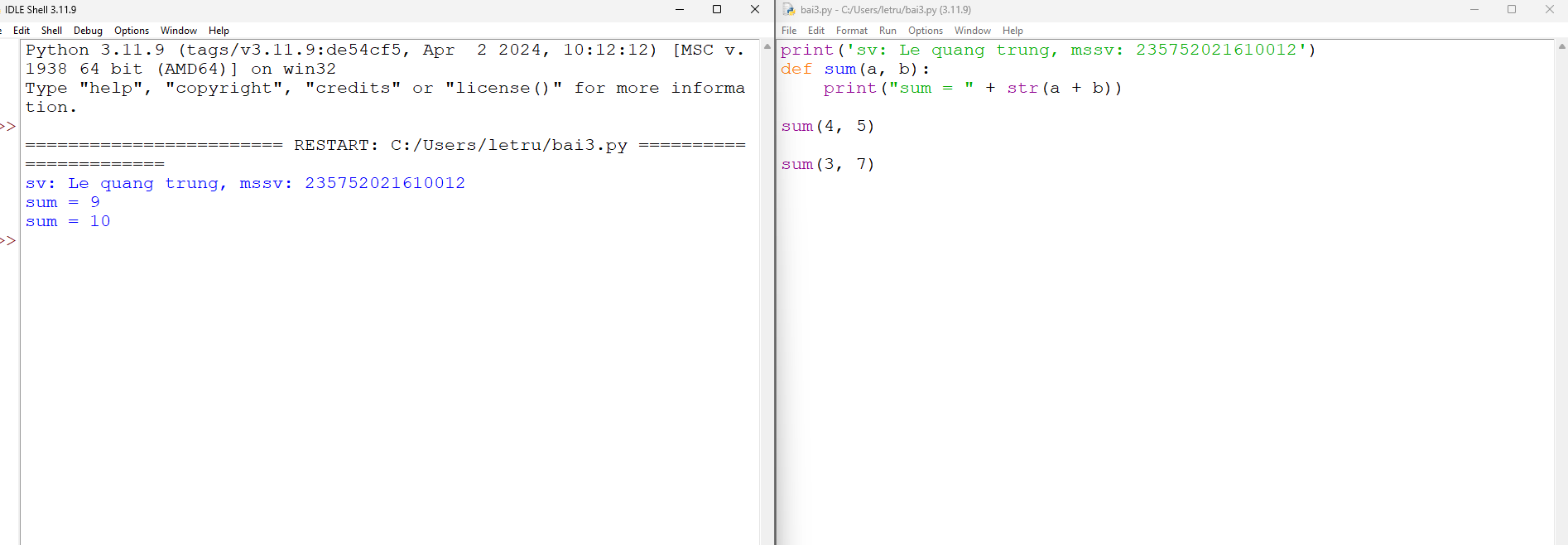
Description automatically generated

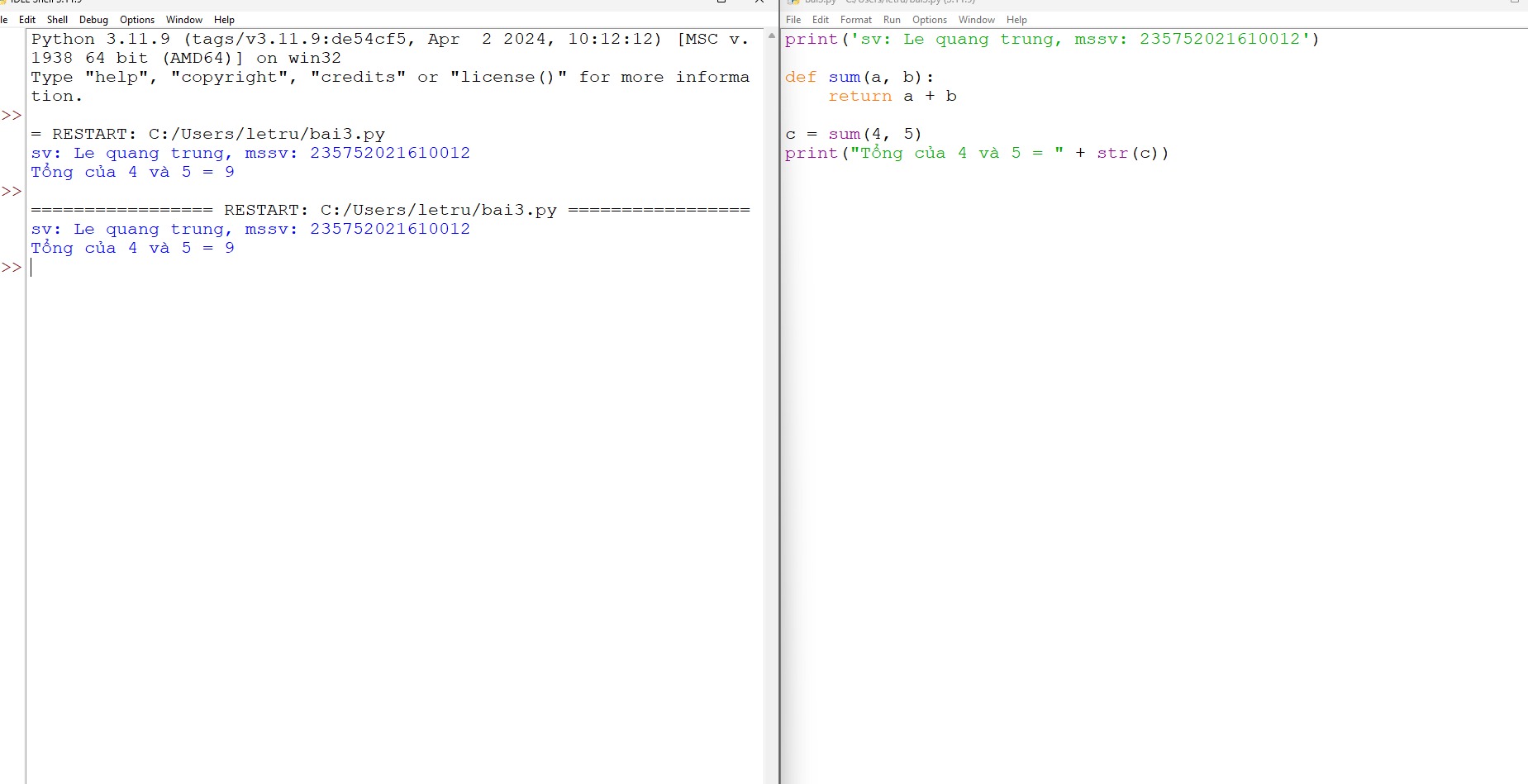
* 1. **Câu hỏi kiểm tra**
  2. **Tài liệu tham khảo**

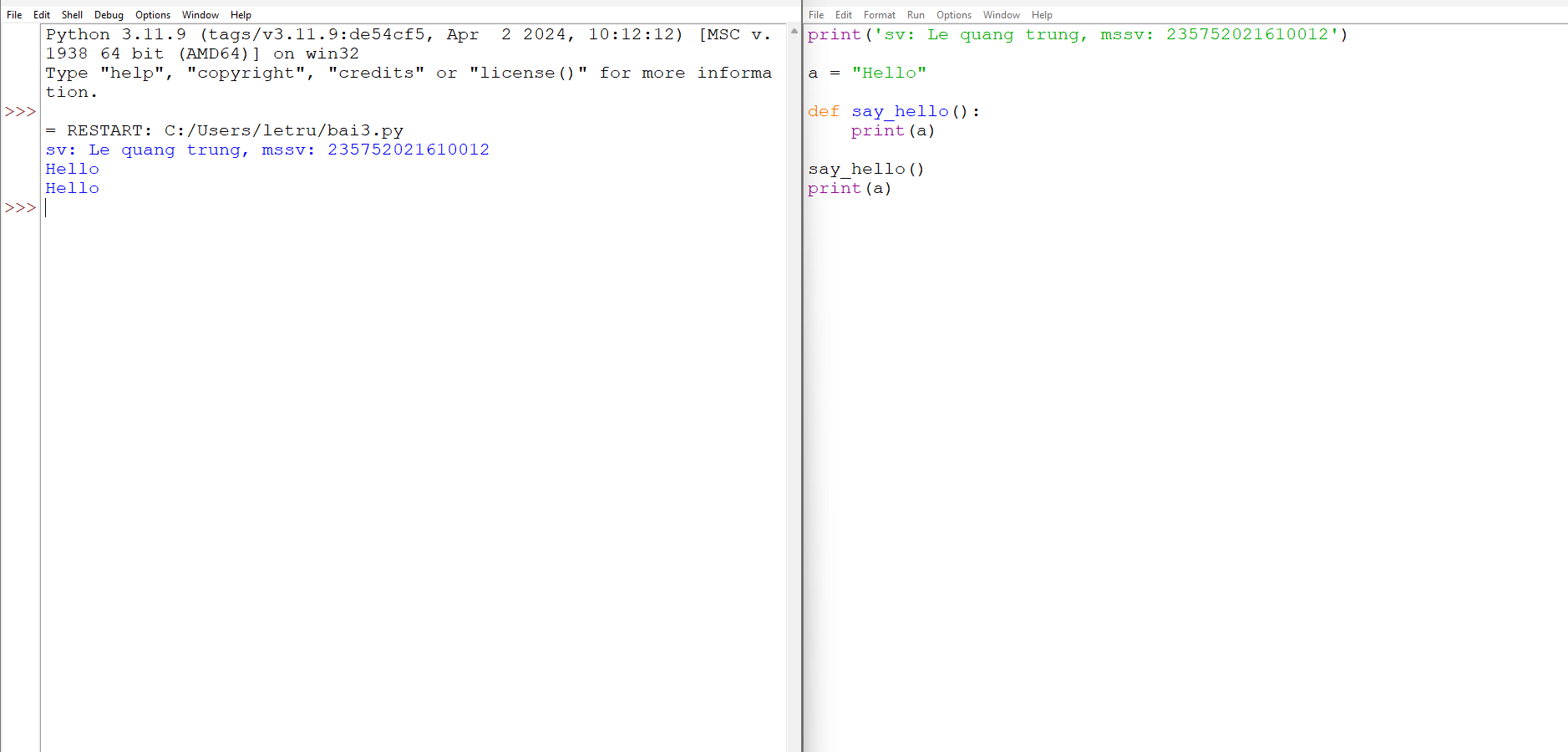
**BÀI 3. Lập trình hàm trong Python**

* 1. **Mục đích**
  2. **Cơ sở lý thuyết**
  3. **Thiết bị thực hành thí nghiệm và vật tư tiêu hao**
  4. **Các bước tiến hành**

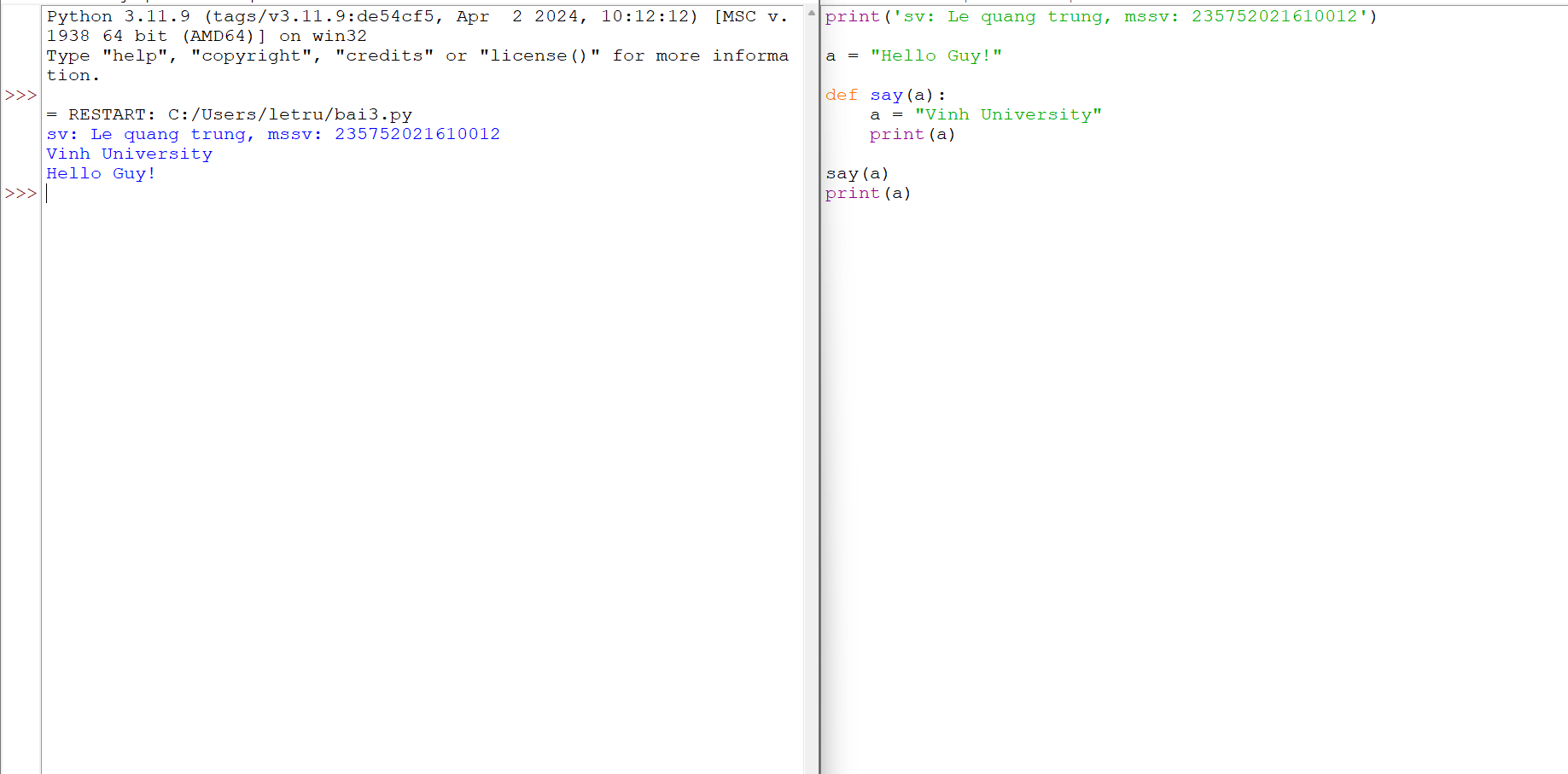
1. **Viết hàm sum() tính tổng hai số**

****

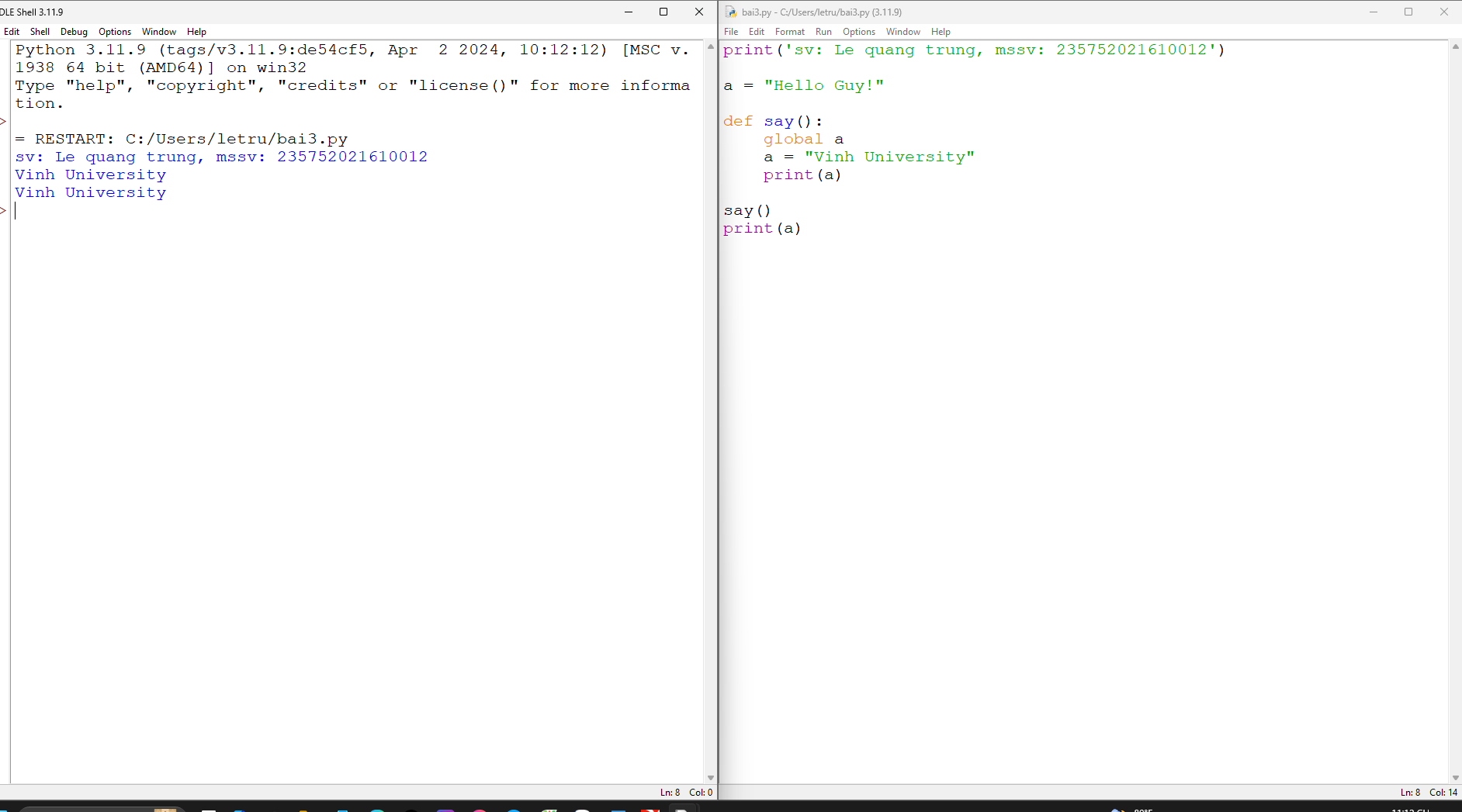
1. **Viết hàm sum() với kết quả trả về**
2. **Tìm và sửa lỗi chương trình**

****

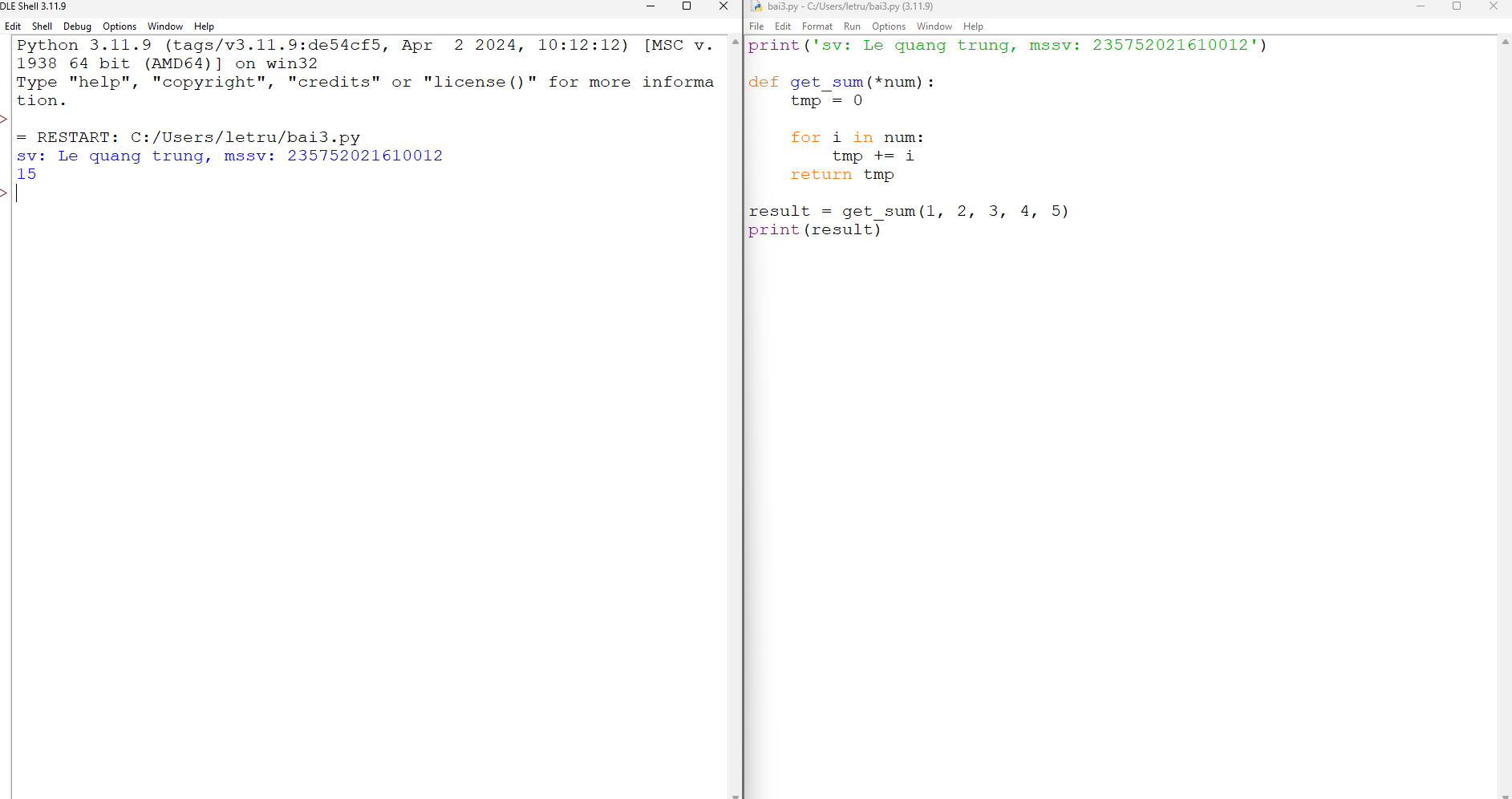
1. **Viết chương trình có phạm vi biến như sau**

****

1. **Viết chương trình sau và xem sự thay đổi của biến**

****

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

****

1. **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**

**A screenshot of a computer

Description automatically generated**

1. **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**

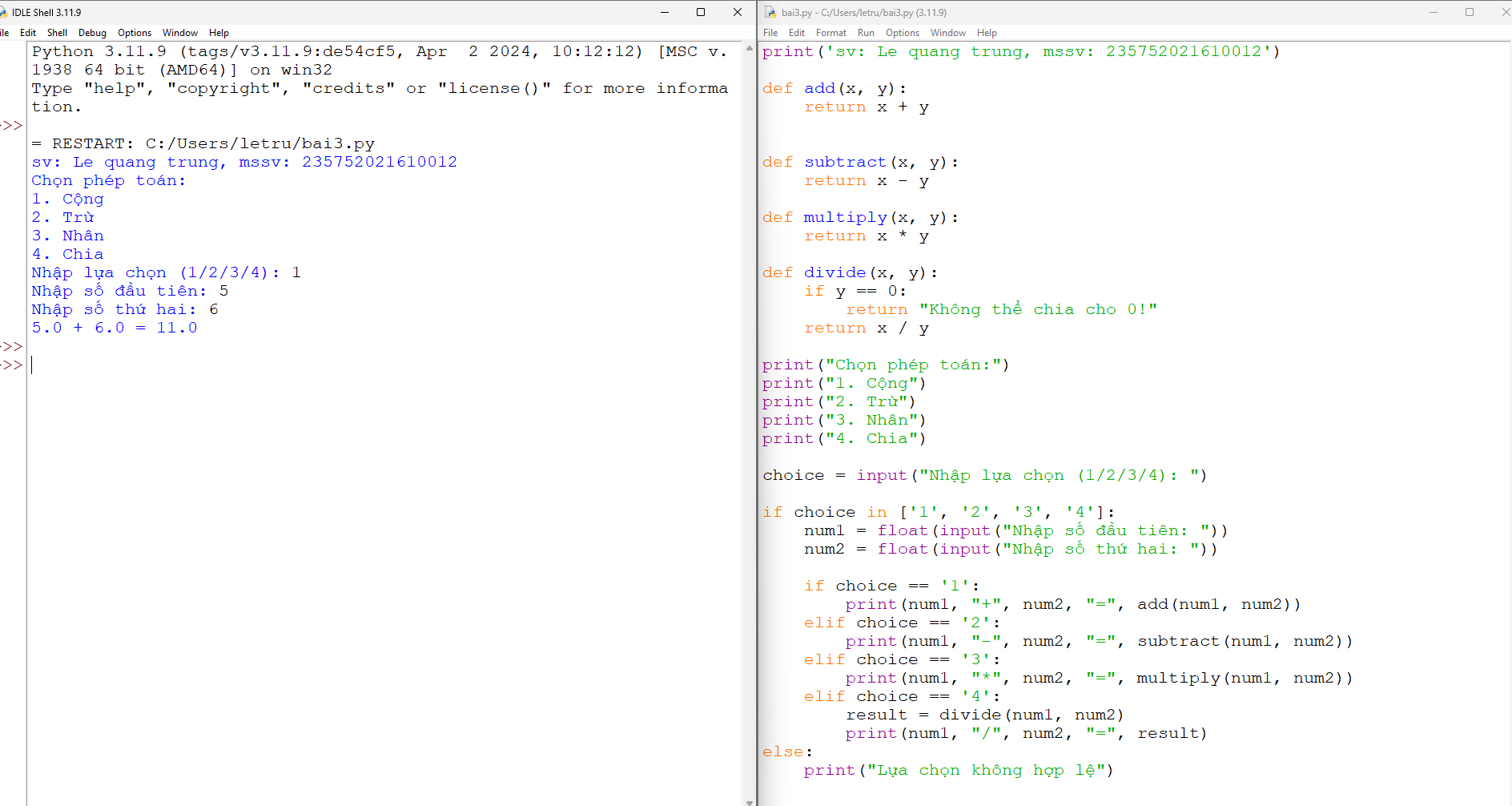
**DOWN 3**

**LEFT 3**

**RIGHT 3**

**Các con số sau phía sau hướng di chuyển chính là số bước đi. Hãy viet 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 só nguyên gần nhất**

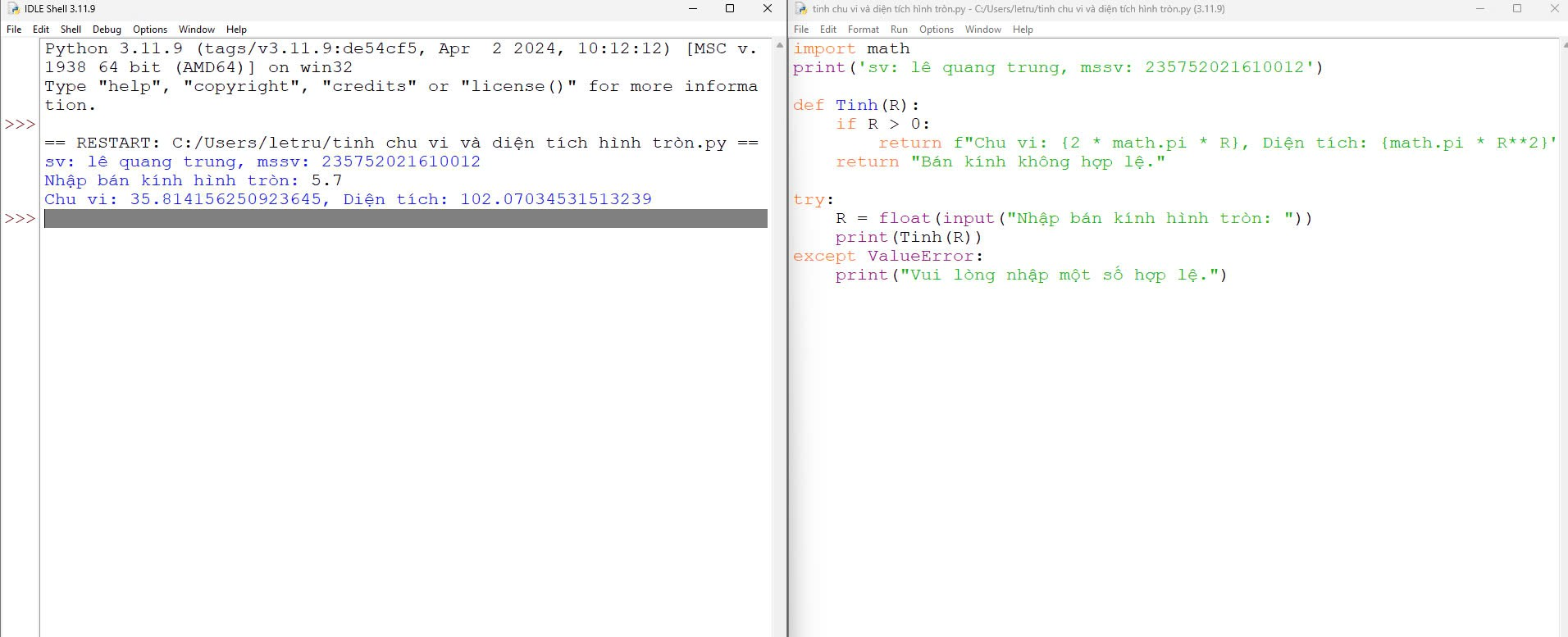
1. **Chương trình máy tính thực hiện các phép tính đơn giản**

****

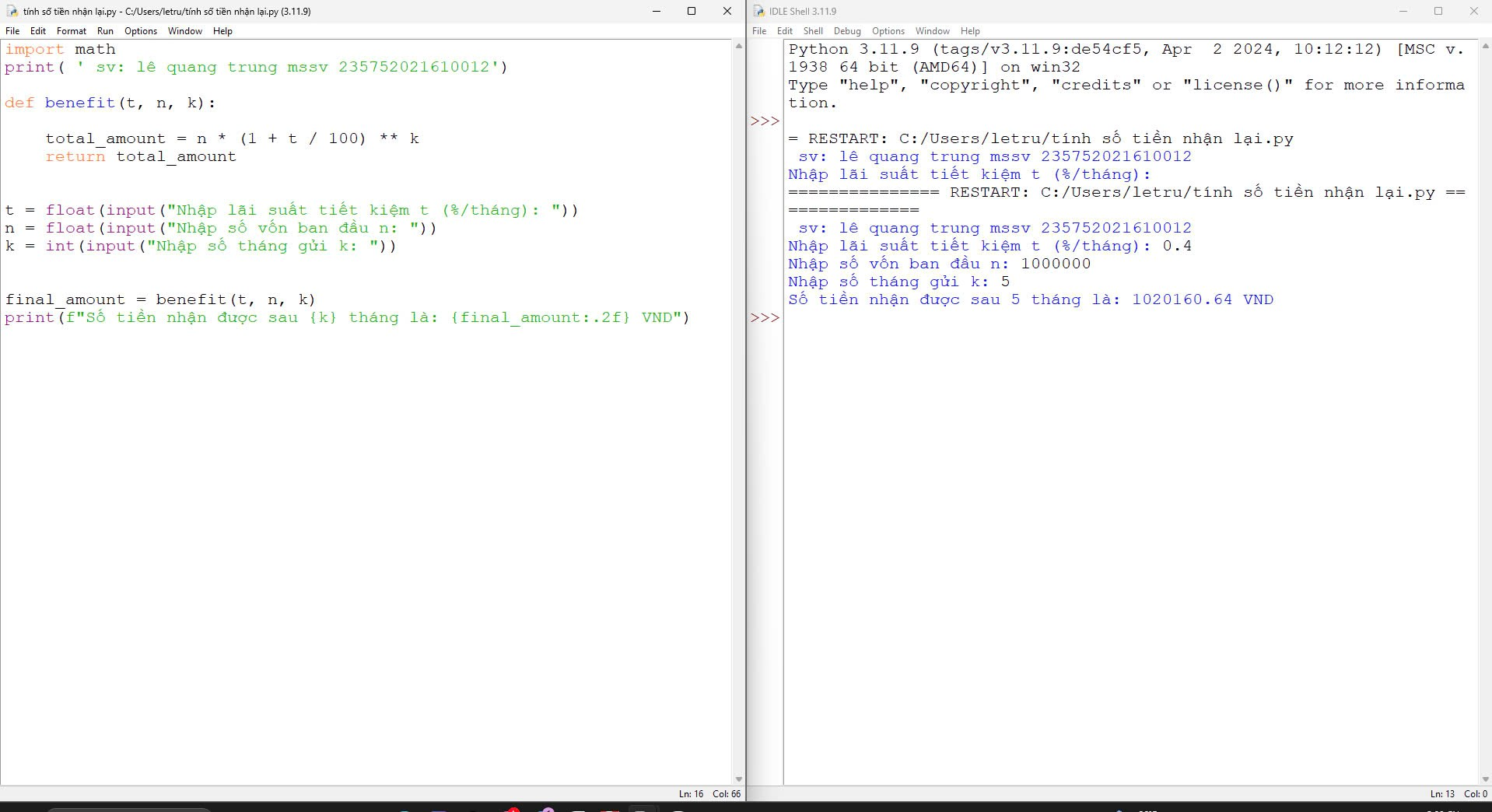
1. **Viết hàm “def Tinh(R):” tính chu vi và diện tích hình tròn, với bán kính R được**

**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**

****

1. **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 hàmdef benefit(t,n,k)**

****

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

* 1. **Mục đích**
  2. **Cơ sở lý thuyết**
  3. **Thiết bị thực hành thí nghiệm và vật tư tiêu hao**
  4. **Các bước tiến hành**

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

**A screenshot of a computer

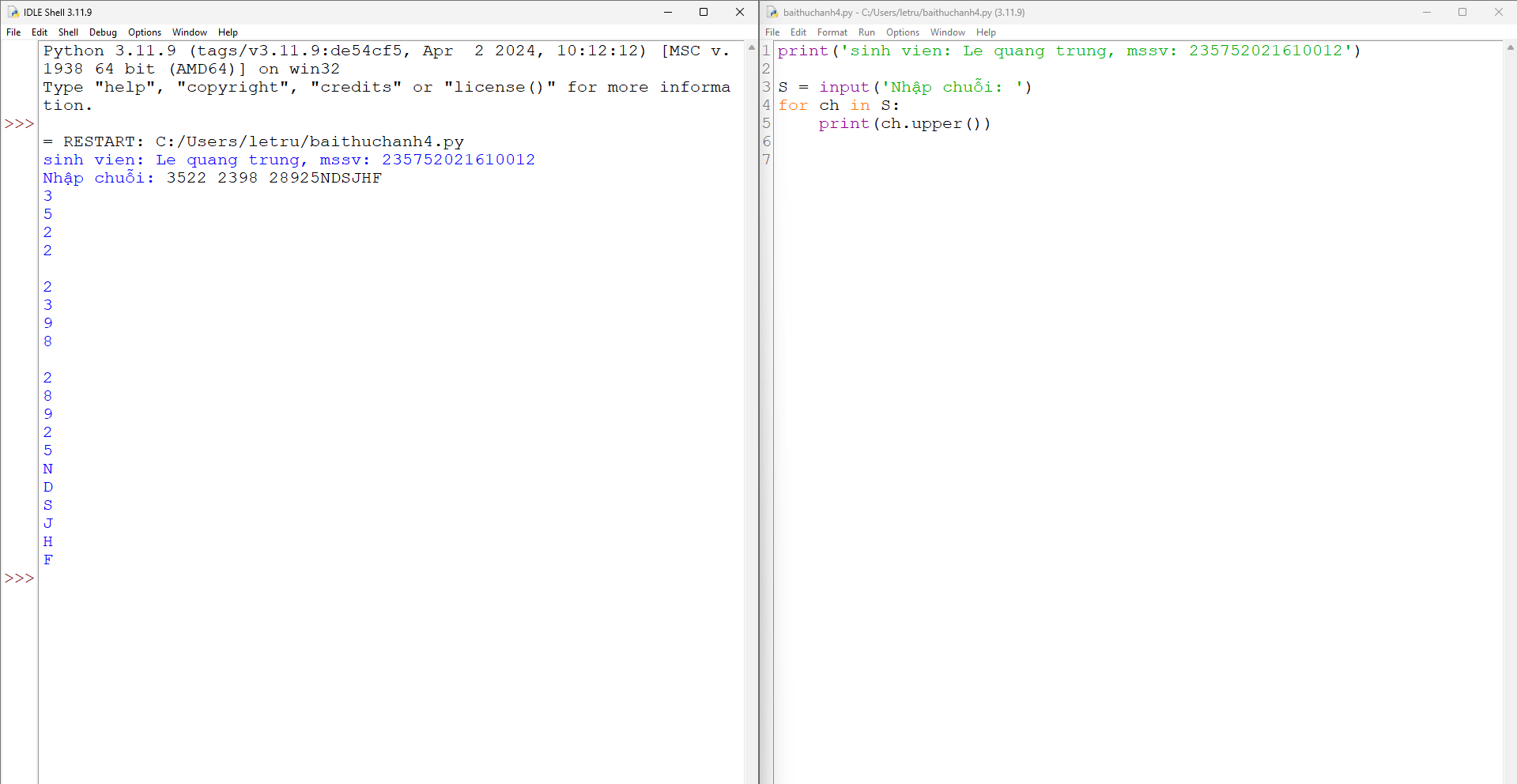
Description automatically generated**

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

**A screenshot of a computer

Description automatically generated**

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

****

1. **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**

**A screenshot of a computer screen

Description automatically generated**

**Giải thích:**

1. **input('Nhập danh sách: '): Nhập một chuỗi từ bàn phím.**
2. **.split(): Phân tách chuỗi thành các phần tử của danh sách, dựa vào dấu trống (space) hoặc tab.**
3. **print(ds): In toàn bộ danh sách.**
4. **for so in ds: print(so): Duyệt qua từng phần tử trong danh sách và in ra từng phần tử trên một dòng riêng.**
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”)**

A screenshot of a computer screen

Description automatically generated

**1. input('Nhập danh sách các từ: ').split(): Nhập chuỗi và tách các từ thành danh sách, sử dụng dấu cách hoặc tab để phân chia.**

**2. ds.reverse(): Đảo ngược thứ tự của các phần tử trong danh sách.**

**3. ' '.join(ds): Nối lại các từ trong danh sách với dấu cách giữa chúng và in ra kết quả.**

1. **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).**

A screenshot of a computer screen

Description automatically generated

**Giải thích:**

1. **input('Nhập chuỗi: '): Nhập một chuỗi từ bàn phím.**
2. **[ch for ch in chuoi if not ch.isdigit()]: Sử dụng list comprehension để tạo ra một danh sách chứa các ký tự không phải là chữ số.**
3. **''.join(...): Nối lại các ký tự trong danh sách thành một chuỗi mới.**
4. **print('Chuỗi mới:', chuoi\_moi): In ra chuỗi mới đã loại bỏ tất cả các chữ số.**
5. **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.**

**A screenshot of a computer

Description automatically generated**

**Giải thích:**

1. **input('Nhập họ và tên: ').split(): Nhập tên từ bàn phím và dùng split() để tách chuỗi thành danh sách dựa trên dấu cách.**
2. **name[0]: Lấy phần tử đầu tiên (phần họ).**
3. **name[1]: Lấy phần tử thứ hai (phần tên).**
4. **Sau đó in ra cả phần họ và tên.**
5. **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**

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **input('Nhập chuỗi: '): Nhập một chuỗi từ bàn phím.**
2. **''.join([ch for ch in chuoi if not ch.isdigit()]): Duyệt qua từng ký tự trong chuỗi và giữ lại những ký tự không phải là chữ số. Phương thức isdigit() trả về True nếu ký tự là số.**
3. **print('Chuỗi mới:', chuoi\_moi): In ra chuỗi mới đã loại bỏ tất cả các chữ số.**
4. **Nhập một list từ bàn phím**

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **input('Nhập các phần tử của list: '): Nhập một chuỗi từ bàn phím, mỗi phần tử cách nhau bởi dấu cách.**
2. **.split(): Tách chuỗi thành một danh sách, các phần tử trong chuỗi được ngăn cách bởi dấu cách hoặc tab.**
3. **print(ds): In danh sách vừa nhập.**
4. **Cắt list: lấy list nhưng bỏ phần tử đầu và cuối**

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **ds[1:-1]: Slicing danh sách, bắt đầu từ chỉ số 1 (phần tử thứ hai) đến chỉ số -1 (phần tử cuối cùng, nhưng không bao gồm nó).**
2. **print(ds\_cut): In ra danh sách đã cắt**
3. **Thêm phần tử vào list**

A screenshot of a computer

Description automatically generated

**Giải thích:**

* **remove(): Loại bỏ phần tử dựa trên giá trị. Nếu không tìm thấy, nó sẽ gây ra lỗi ValueError.**
* **pop(): Loại bỏ phần tử tại vị trí đã chỉ định và trả về nó. Nếu không chỉ định, mặc định sẽ loại bỏ phần tử cuối cùng.**
* **del: Loại bỏ phần tử tại chỉ số nhất định mà không trả về giá trị.**

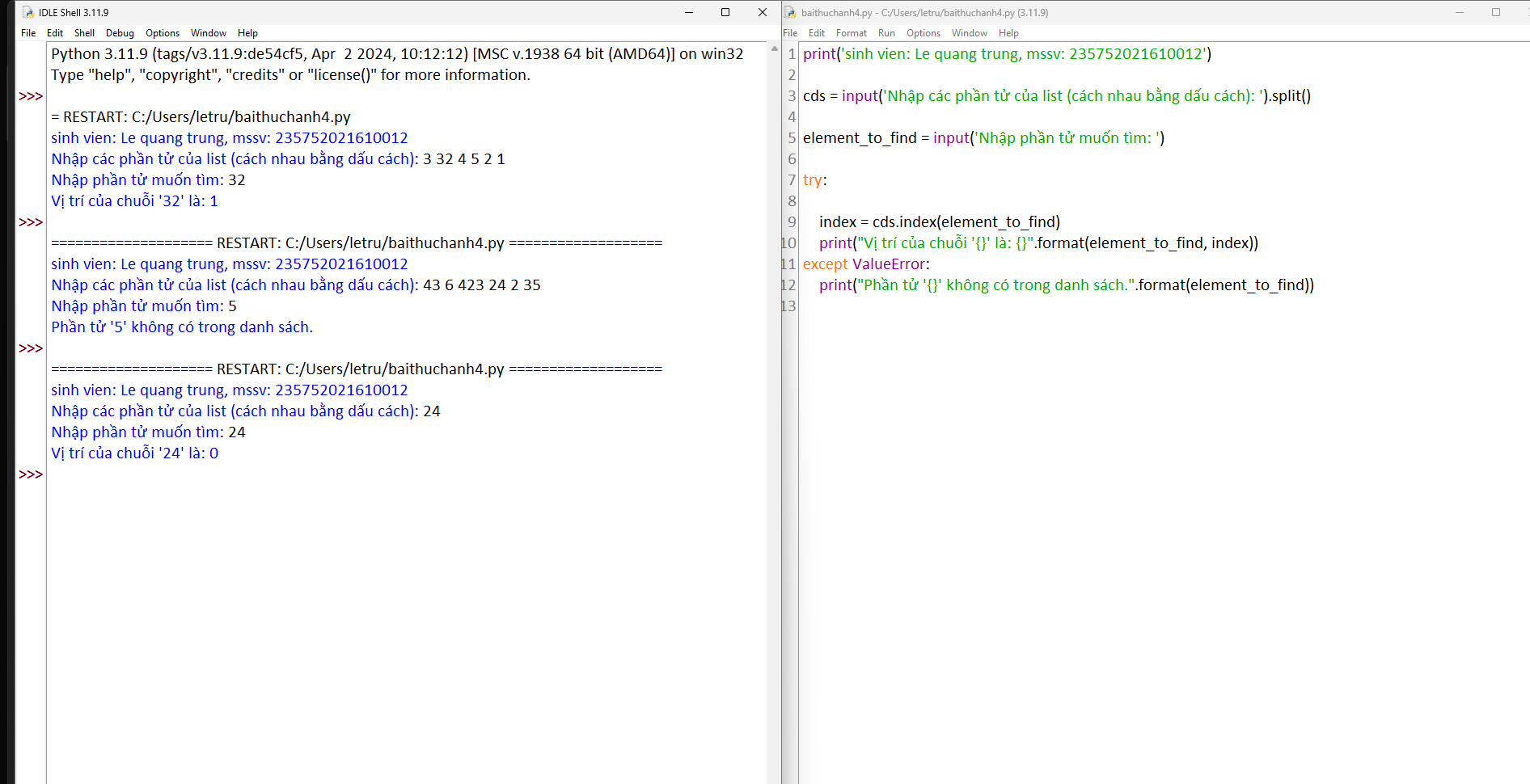
1. **Bỏ phần tử khỏi list**

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **cds.index(element\_to\_find): Tìm vị trí đầu tiên của phần tử element\_to\_find trong danh sách cds. Nếu không tìm thấy, nó sẽ gây ra lỗi ValueError.**
2. **try và except: Bằng cách sử dụng khối try, bạn có thể xử lý lỗi nếu phần tử không tồn tại trong danh sách, thay vì để chương trình bị lỗi.**
3. **Tìm kiếm phần tử trong list**



**Giải thích:**

1. **cds.index(element\_to\_find): Tìm vị trí đầu tiên của phần tử element\_to\_find trong danh sách cds. Nếu không tìm thấy, nó sẽ gây ra lỗi ValueError.**
2. **try và except: Bằng cách sử dụng khối try, bạn có thể xử lý lỗi nếu phần tử không tồn tại trong danh sách, thay vì để chương trình bị lỗi.**

**14. Sắp xếp các phần tử trong list**

**A screenshot of a computer

Description automatically generated**

**Giải thích:**

1. **ds.sort(): Sắp xếp các phần tử trong danh sách ds theo thứ tự tăng dần.**
2. **Vòng lặp for ch in ds:: Duyệt qua từng phần tử trong danh sách đã sắp xếp và in ra từng phần tử.**
3. **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**

A screenshot of a computer

Description automatically generated

**Giải thích:**

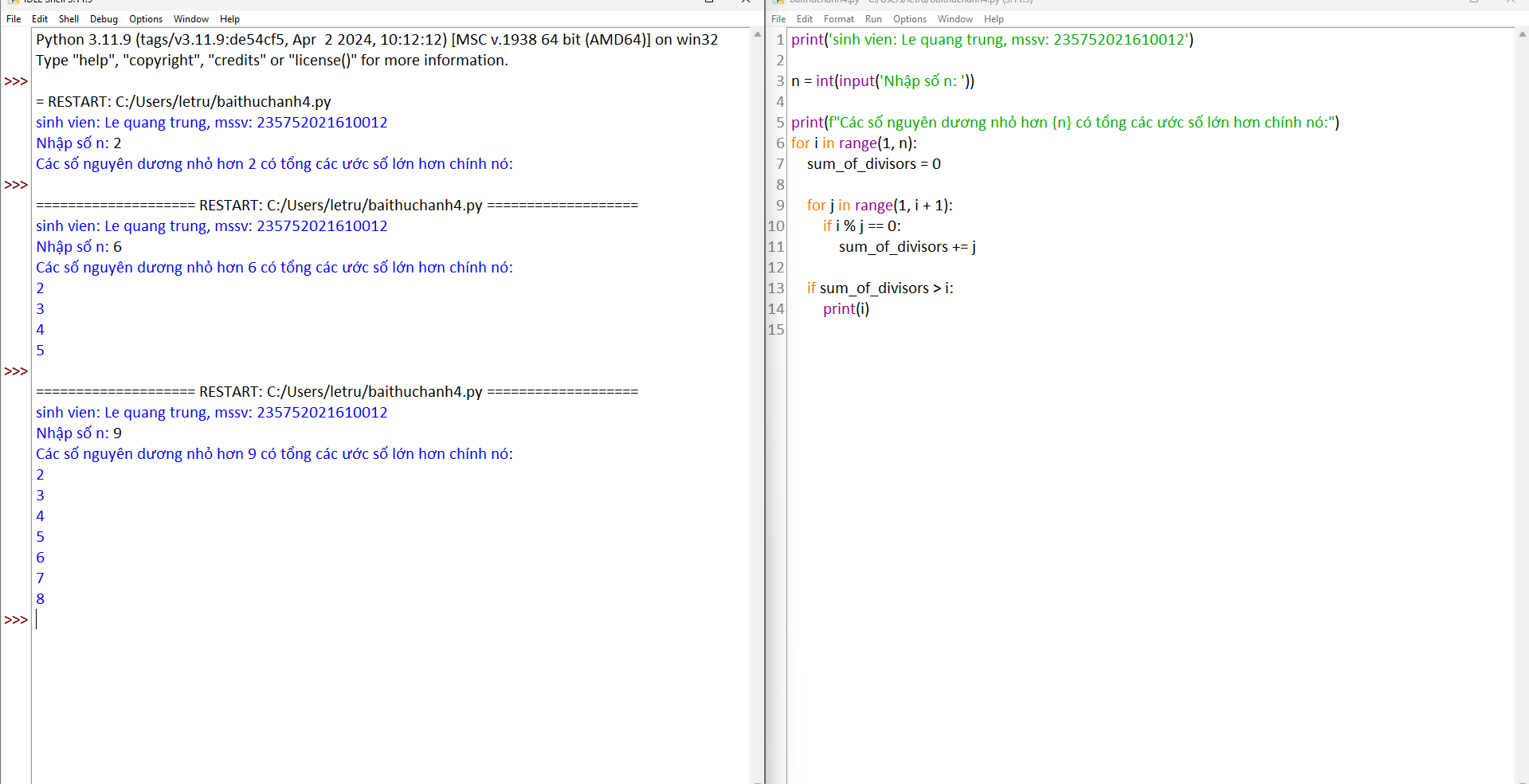
1. **input('Nhập các từ tiếng Anh (cách nhau bằng dấu cách): '): Nhập chuỗi từ bàn phím.**
2. **input\_string.split(): Tách chuỗi thành danh sách các từ, sử dụng dấu cách làm ký tự phân cách.**
3. **words.sort(): Sắp xếp danh sách words theo thứ tự từ điển.**
4. **Vòng lặp for word in words:: Duyệt qua từng từ trong danh sách đã sắp xếp và in ra.**
5. **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.**

**A screenshot of a computer screen

Description automatically generated**

**Giải thích:**

1. **input('Nhập chuỗi các số nhị phân (cách nhau bằng dấu phẩy): '): Nhập chuỗi từ bàn phím.**
2. **input\_string.split(','): Tách chuỗi thành danh sách các số nhị phân, sử dụng dấu phẩy làm ký tự phân cách.**
3. **Vòng lặp for value in binary\_values:: Duyệt qua từng giá trị trong danh sách đã tách và in ra.**
4. **value.strip(): Loại bỏ khoảng trắng ở đầu và cuối của mỗi giá trị, nếu có**
5. **Nhập số n, in ra màn hình các số nguyên dương nhỏ hơn n có tổng các ước số lớn hơn chính nó.**

****

**Giải thích:**

1. **int(input('Nhập số n: ')): Nhập số nguyên dương nnn từ bàn phím.**
2. **Vòng lặp for i in range(1, n): Duyệt qua các số từ 1 đến n−1n-1n−1.**
3. **Vòng lặp for j in range(1, i + 1): Tìm các ước số của iii.**
4. **if i % j == 0: Kiểm tra xem jjj có phải là ước số của iii không.**
5. **sum\_of\_divisors > i: So sánh tổng các ước số với iii và in ra nếu điều kiện thỏa mãn.**
6. **Hãy nhập số nguyên n, tạo một list gồm các số fibonacci nhỏ hơn n và in ra màn hình.**

**A screenshot of a computer

Description automatically generated**

**Giải thích:**

1. **int(input('Nhập số nguyên n: ')): Nhập số nguyên nnn từ bàn phím.**
2. **fibonacci\_list = []: Khởi tạo một danh sách rỗng để chứa các số Fibonacci.**
3. **a, b = 0, 1: Khởi tạo hai số Fibonacci đầu tiên, 0 và 1.**
4. **while a < n:: Vòng lặp sẽ tiếp tục cho đến khi số Fibonacci aaa lớn hơn hoặc bằng nnn.**
5. **fibonacci\_list.append(a): Thêm số Fibonacci aaa vào danh sách.**
6. **a, b = b, a + b: Cập nhật aaa và bbb cho lần lặp tiếp theo**
7. **Hãy tạo ra tuple P gồm các số nguyên tố nhỏ hơn 1 triệu.**

**A screenshot of a computer

Description automatically generated**

**Giải thích:**

1. **sieve\_of\_eratosthenes(limit): Hàm này nhận vào một giới hạn và trả về một tuple chứa các số nguyên tố nhỏ hơn giới hạn đó.**
2. **is\_prime = [True] \* limit: Khởi tạo một danh sách để đánh dấu số nguyên tố, bắt đầu với tất cả các số đều được đánh dấu là số nguyên tố.**
3. **is\_prime[0] = is\_prime[1] = False: Đánh dấu 0 và 1 không phải là số nguyên tố.**
4. **Vòng lặp for num in range(2, int(limit\*\*0.5) + 1): Duyệt qua các số từ 2 đến căn bậc hai của giới hạn.**
5. **for multiple in range(num\*num, limit, num): Đánh dấu tất cả các bội số của num là không nguyên tố.**
6. **primes = [num for num, prime in enumerate(is\_prime) if prime]: Tạo danh sách các số nguyên tố từ danh sách đánh dấu.**
7. **return tuple(primes): Trả về một tuple chứa các số nguyên tố.**
8. **Nhập n, in n dòng đầu tiên của tam giác pascal.**

**A screenshot of a computer

Description automatically generated**

**Giải thích:**

1. **def print\_pascals\_triangle(n):: Định nghĩa hàm để in n dòng của tam giác Pascal.**
2. **for i in range(n):: Duyệt qua từng dòng từ 0 đến n-1.**
3. **value = 1: Khởi tạo giá trị đầu dòng.**
4. **for j in range(i + 1):: Duyệt qua các giá trị trong dòng i.**
5. **print(value, end=' '): In giá trị hiện tại.**
6. **value = value \* (i - j) // (j + 1): Cập nhật giá trị cho phần tử tiếp theo trong dòng theo công thức.**
7. **print(): In một dòng trống sau khi in tất cả các giá trị của một dòng**
8. **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 sốchia hết cho 5 thành dãy phân tách bởi dấu phẩy.**

**Ví dụ đầu vào là: 0100,0011,1010,1001**

**Đầu ra sẽ là: 1010**

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **Nhập đầu vào: Chuỗi các số nhị phân được nhập và phân tách thành danh sách.**
2. **Kiểm tra chia hết cho 5: Dùng hàm int(binary, 2) để chuyển số nhị phân sang hệ thập phân, sau đó kiểm tra xem có chia hết cho 5 không.**
3. **Kết quả: In ra các số nhị phân chia hết cho 5 dưới dạng chuỗi phân tách bởi dấu phẩy**
4. **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**

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **Nhập đầu vào: Chuỗi câu được nhập từ người dùng.**
2. **Duyệt từng ký tự: Sử dụng vòng lặp để kiểm tra từng ký tự:**
   * **isalpha() để kiểm tra chữ cái.**
   * **isdigit() để kiểm tra chữ số.**
3. **In kết quả: Hiển thị số lượng chữ cái và chữ số**
4. **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**

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **Nhập đầu vào: Chuỗi câu được nhập từ người dùng.**
2. **Duyệt từng ký tự: Sử dụng vòng lặp để kiểm tra từng ký tự:**
   * **isupper() để kiểm tra chữ hoa.**
   * **islower() để kiểm tra chữ thường.**
3. **In kết quả: Hiển thị số lượng chữ hoa và chữ thường.**
4. **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**

A screenshot of a computer

Description automatically generated

**Giải thích:**

1. **Nhập đầu vào: Chuỗi số từ người dùng, phân tách bởi dấu phẩy.**
2. **Chuyển sang danh sách số nguyên: Dùng split() để tách các số và chuyển thành kiểu int.**
3. **Lọc số lẻ: Duyệt qua danh sách và chọn các số lẻ (số chia 2 dư 1).**
4. **In kết quả: In các số lẻ dưới dạng chuỗi, phân tách bởi dấu phẩy.**
5. **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**

**A screenshot of a computer

Description automatically generated**

**Giải thích:**

1. **Khởi tạo số tiền: Đặt biến balance bằng 0 để lưu trữ số tiền còn lại trong tài khoản.**
2. **Nhập nhật ký giao dịch: Dùng vòng while để nhận nhiều giao dịch từ người dùng.**
3. **Phân tích giao dịch:**
   * **Dùng split() để tách loại giao dịch (D hoặc W) và số tiền.**
   * **Cộng vào balance nếu là D (gửi), hoặc trừ nếu là W (rút).**
4. **Hiển thị kết quả: In số dư còn lại trong tài khoản**

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

**1.1. 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.**

**1.2. 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.**

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

**Thiết bị thực hành thí nghiệm: Máy tính**

**Vật tư tiêu hao: Không**

**1.4. Các bước tiến hành**

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

* 1. **Sử dụng module. Định nghĩa một module toán học gọi là mymath và sử dụng module này từ một tập lệnh riêng biệt.**
  2. **Sử dụng thư viện tiêu chuẩn của python (datetime)**
  3. **Viết chương trình sử dụng thư viện NumPy để tạo một mảng với các giá trị nằm trong khoảng từ 12 đến 38**

**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]**

* 1. **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)**

**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:**

* 1. **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**

**A screenshot of a computer

Description automatically generated**

**Giải thích**

1. **mymodule.py:**
   * **sort\_list(lst): Sắp xếp danh sách theo thứ tự tăng dần.**
   * **find\_max(lst): Tìm phần tử lớn nhất.**
   * **find\_min(lst): Tìm phần tử nhỏ nhất.**
2. **main.py:**
   * **Nhập số lượng và giá trị phần tử từ bàn phím.**
   * **Gọi các hàm từ module mymodule để xử lý danh sách.**
   * **In ra danh sách đã sắp xếp, phần tử lớn nhất và nhỏ nhất.**
   1. **In ra vị trí phần tử lớn nhất và nhỏ nhất tìm được ở bài tập trên**

**A screenshot of a computer

Description automatically generated**

**Giải thích:**

1. **mymodule.py:**
   * **find\_max\_index(lst): Tìm vị trí của phần tử lớn nhất bằng cách sử dụng lst.index(max\_value).**
   * **find\_min\_index(lst): Tìm vị trí của phần tử nhỏ nhất bằng lst.index(min\_value).**
2. **main.py:**
   * **Gọi các hàm từ module mymodule để tìm và in ra vị trí của phần tử lớn nhất và nhỏ nhất.**
   1. **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.**