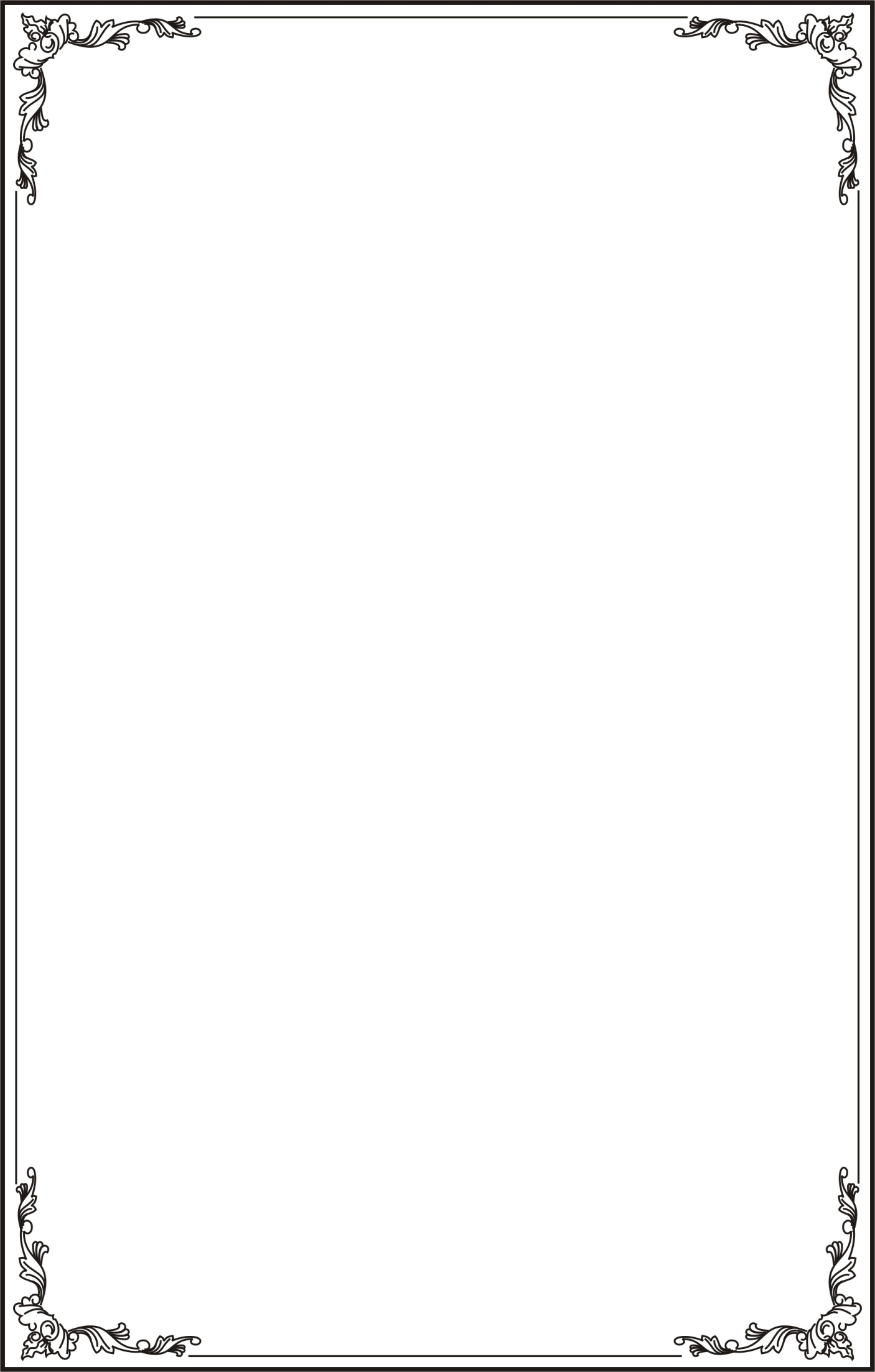
****

# TRƯỜNG ĐẠI HỌC VINH

**VIỆN KỸ THUẬT VÀ CÔNG NGHỆ**



**BÁO CÁO THỰC HÀNH**

**Ngành kỹ thuật điều khiển và tự động hóa**

**Môn: kỹ thuật lập trình**

**Họ và Tên : Trần Đức Hiếu**

**Msv : 245752021610073**

**GVHD: MAI THẾ ANH**

Mục lục

[Bài 1. Thực hiện các thuật toán bằng phần mềm Flowgorithm 2](#_bookmark0)

[Bài 2. Các cú pháp, kiểu dữ liệu, lệnh điều khiển trong lập trình Python](#_bookmark1) 7

Bài 3, Lập trình hàm trong Python 21

Bài 4. Các kiểu dữ liệu có cấu trúc trong Python 33

Bài 5. Thiết kế module trong Python

Bài 6.Lập trình đối tượng trong Python

Bài 7. Thao tác trên tập tin và thư mục Python

Bài 8. Lập trình giao diện trong Python .................................................................

**Chương 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.

- Trực quan hóa các bước thực hiện thay vì chỉ đọc code.

- Hỗ trợ học thuật toán và lập trình cho người mới bắt đầu.

- Có thể mô phỏng và chạy thử từng bước để kiểm tra đúng sai.

Làm quen với tư duy lập trình có cấu trúc (rẽ nhánh, lặp, xử lý dữ liệu).

**2 các bước thực hiện kết quả**

Viết chương trình nhập vào cạnh và chiều cao tương ứng của một tam giác và in ra màn hình diện tích tam giác.

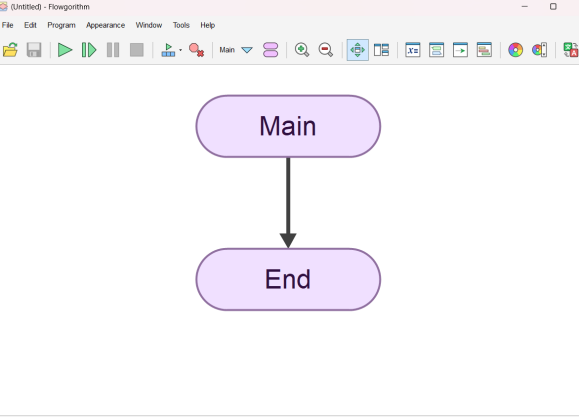
**Bước 1**: xác định đầu vào, đầu ra của bài toán và công thức sử dụng.

- Theo đề bài, ta thấy đầu vào chính là cạnh(a) và chiều cao(h)

- Công thức tính diện tích hình tam giác là: S=1/2\*a\*h

- Đầu ra sẽ là diện tích S

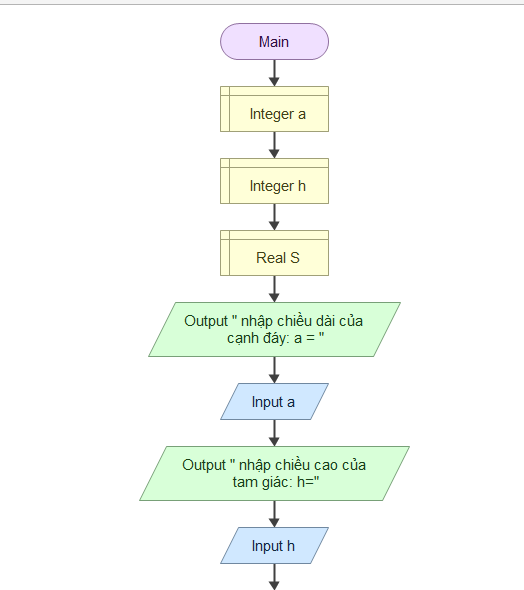
**Bước 2:** Mở Flowgorithm – Khởi chạy file mới.



*Hình 1.1.Giao diện khi bắt đầu.*

*-* Trước tiên ta sẽ dùng Declare để khai báo thuộc tính cho biến a(cạnh), h(chiều cao) và S(diện tích)

- Sau khi khai báo xong ta sẽ dùng output và input để lần lượt hiển thị giải thích câu lệnh và nhập vào biến

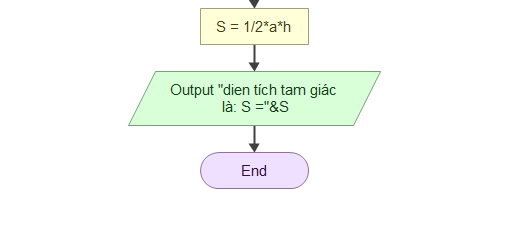


*Hình 1.2. Đầu vào của bước 1*

**Bước 3:** Nhập công thức để tính toán và in ra kết quả

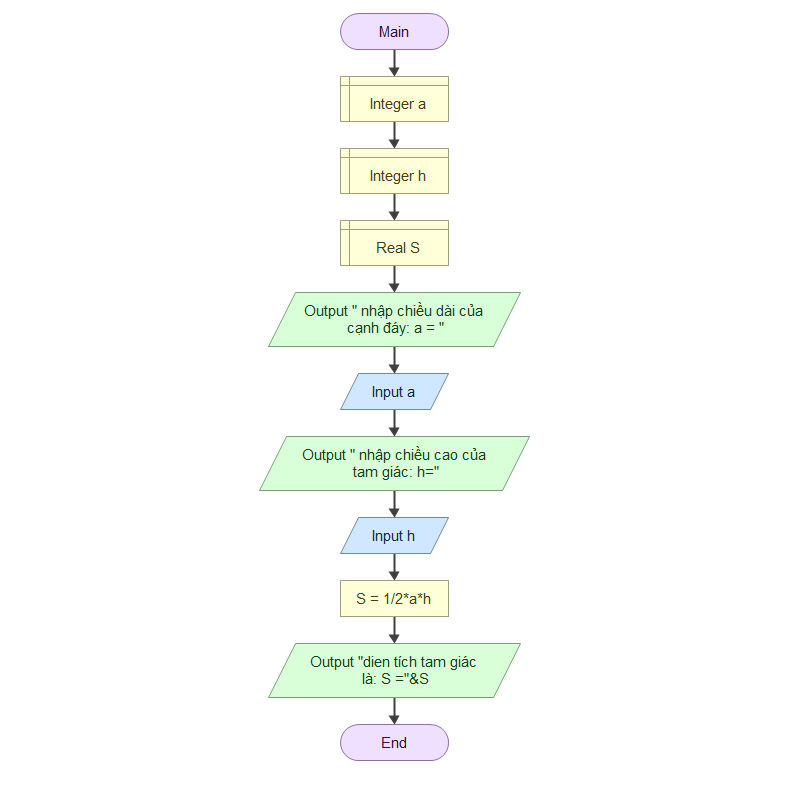
- Điền công thức tính diện tích tam giác: S=1/2\*a\*h

- Dùng lệnh output để in ra kết quả



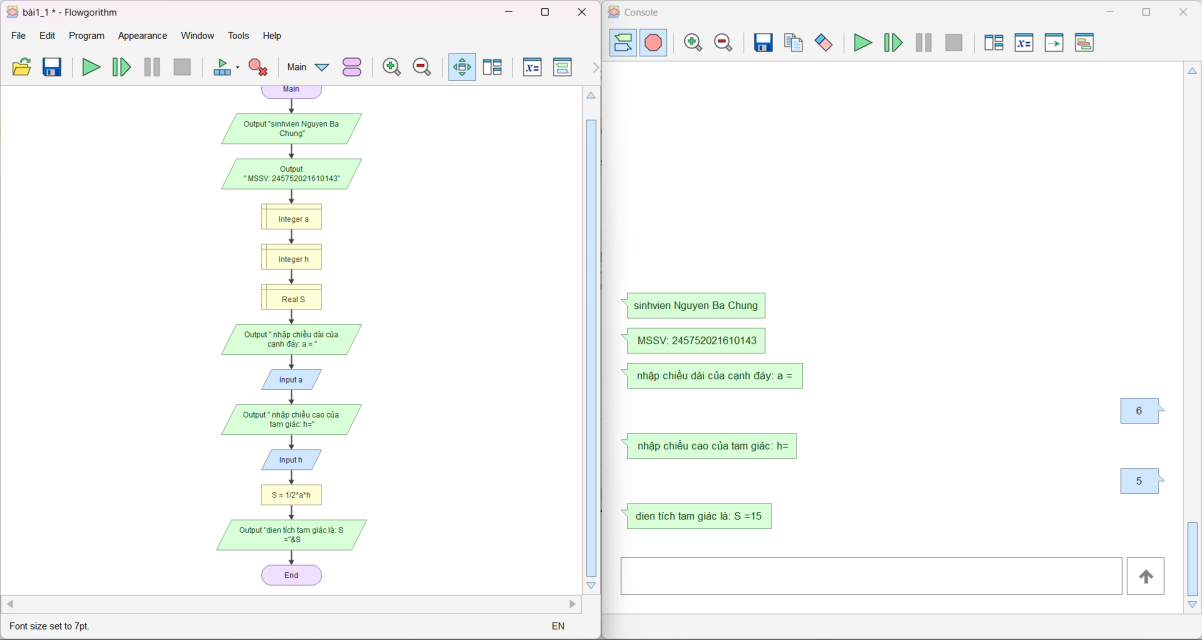
*Hình 1.3. Kết thuc bước 2*

Bước 4: Khởi chạy chương trình ta dùng F5 hoặc Run



*Hình 1.4. Chương trình hoàn thiện*

Sau khi khởi chạy, ta sẽ nhập đầu vào cạnh a = 6 và chiều cao h = 5 ta sẽ được kết quả:

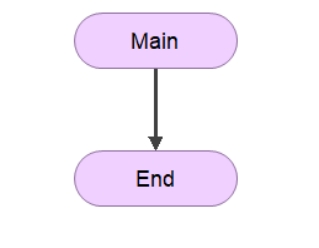


*Hình 1.5. Kết quả chạy chương trình*

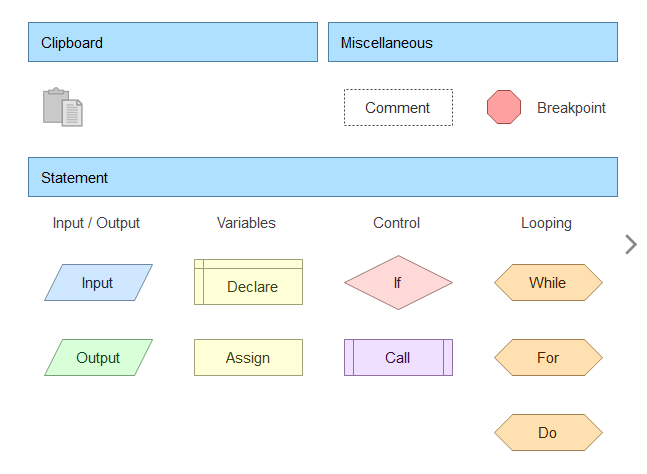
Xây dựng thuật toán kiểm tra số nguyên tố.

- Khi bắt đầu chúng ta ấn chuột phải để sử dụng lệnh:

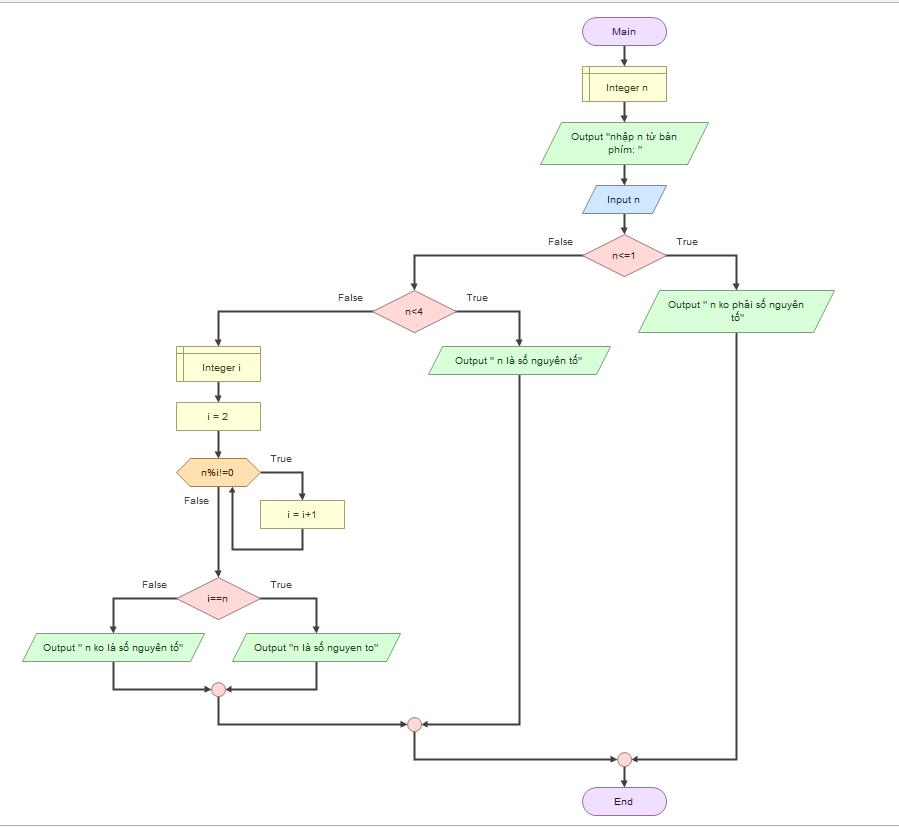
đầu chúng ta ấn chuột phải để sử dụng lệnh:



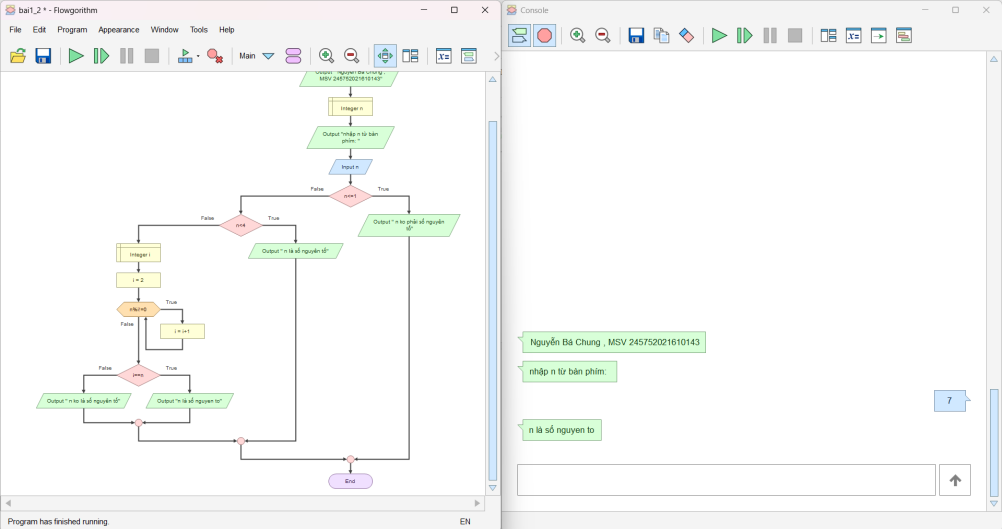
- Tiếp tục ta sử dụng các lệnh integer, input, output, các phép toán so sánh, if else, vòng lặp while để tạo ra các câu lệnh giúp phân biệt được số nguyên tố:



- Khởi chạy chương trình đã thiết kế:



- Kết quả



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

**1 Mục đích.**

- Giúp sinh viên nắm vững các cú pháp cơ bản trong Python như: khai báo biến, kiểu dữ liệu, nhập – xuất dữ liệu.

- Hiểu và vận dụng các lệnh điều khiển: rẽ nhánh (if, if-else), lặp (for, while).

- Biết phát hiện và sửa lỗi cú pháp trong chương trình Python.

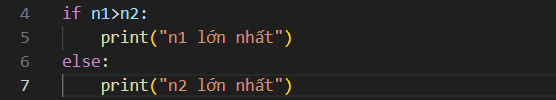
- Rèn luyện kỹ năng viết chương trình hoàn chỉnh, có kết quả đúng logic.

**2 Các bước thực hiện.**

***Bài 2.1 viết đoạn chương trình sau và sửa lỗi.***

- Bước 1: khai báo, nhập dữ liệu.

- Bước 2: sử dụng if - else để so sánh n1, n2

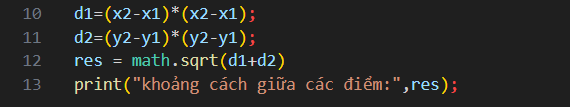


- Bước 3: cho chương trình khởi chạy, in ra kết quả

***Bài 2: viết chương trình nhập 2 điểm và tính khoảng cách.***

- Bước 1: nhập dữ liệu

- Bước 2: sử dụng thuật toán công thức để tính toán.

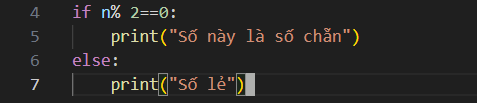


- Bước 3: khởi chạy chương trình và in ra kết quả.

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

- Bước 1: nhập dữ liệu.

- Bước 2: Để biết được đó là số chẵn hay lẻ ta dùng công thức chia hết cho 2



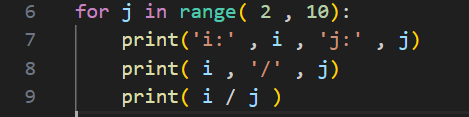
- Bước 3: khởi chạy chương trình và in ra kết quả.

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

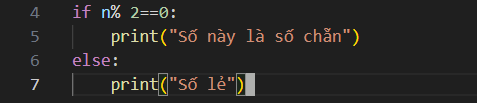
- Bước 1: khởi tạo biến i ban đầu bằng 1.

Screenshot 2025-10-05 084428

- Bước 2: in giá trị i và thực hiện phép chia.



- Bước 3: khởi chạy chương trình và in ra kết quả.



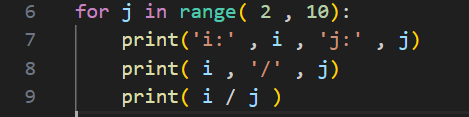
- Bước 3: khởi chạy chương trình và in ra kết quả.

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

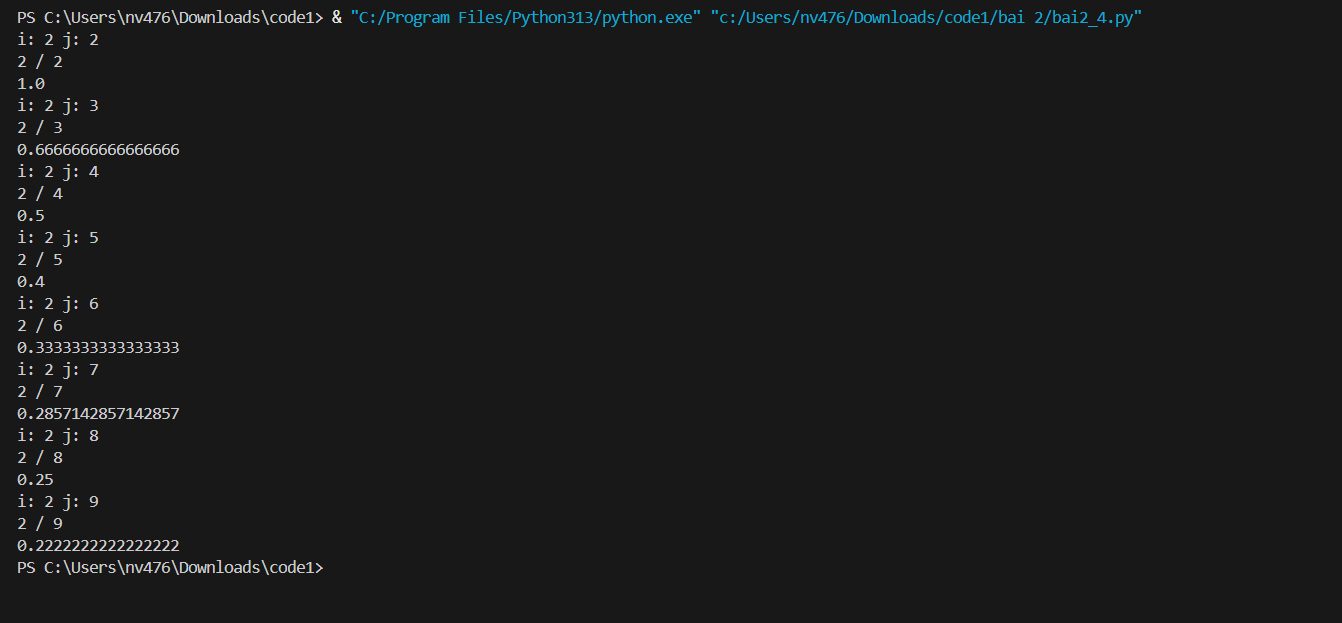
- Bước 1: khởi tạo biến i ban đầu bằng 1.

Screenshot 2025-10-05 084428

- Bước 2: in giá trị i và thực hiện phép chia.

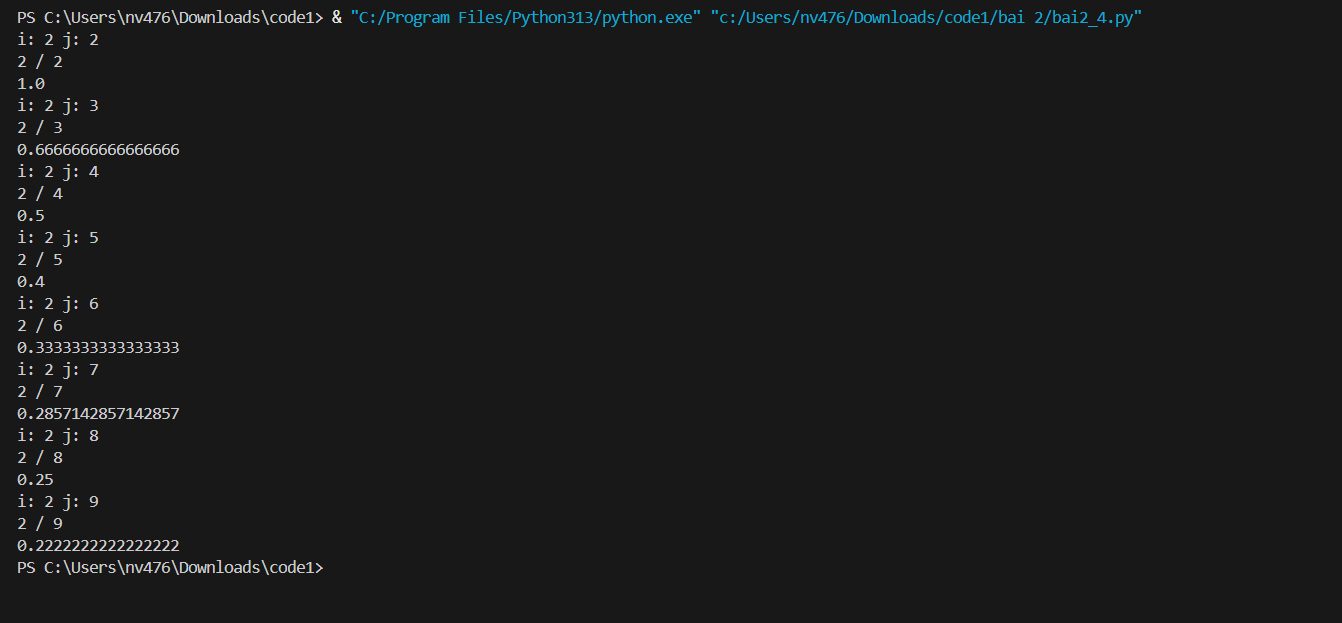


- Bước 3: khởi chạy chương trình và in ra kết quả.



***Bài 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.***

- Bước 1: nhập dữ liệu đầu vào.





- Bước 2: ta sử dụng vòng lặp while.



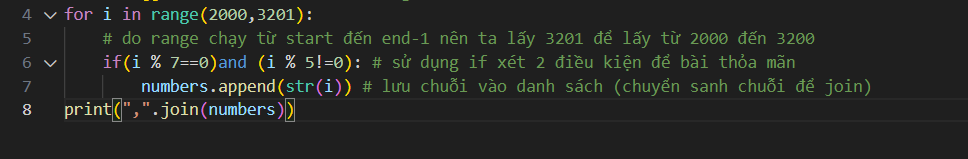
- Bước 3: khởi chạy chương trình và in ra kết quả.

***Bài 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.***

- Bước 1: khai báo numbers (danh sách rỗng để chứa).



- Bước 2: ta sử dụng vòng lặp while.



- Bước 3: chạy chương trình và in ra kết quả.

***Bài 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.***

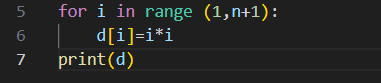
- Bước 1: nhập dữ liệu đầu vào.



- Bước 2: tạo một từ điển rỗng ( dictionary) có tên là d.



- Bước 3: dùng lệnh for chạy qua từ 1 đến n để giải.

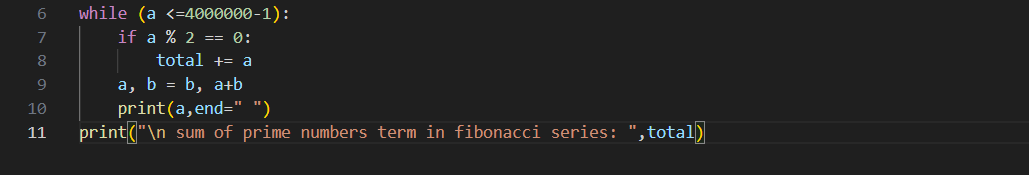


- Bước 4: chạy chương trình và in ra kết quả.

***Bài 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.***

- Bước 1: Khai báo 2 số fibonacci đầu tiên và khai báo tổng ban đầu.

- Bước 2: sử dụng vòng lặp while.

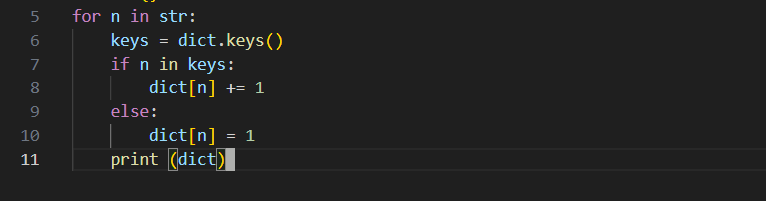


- Bước 3: chạy chương trình và in ra kết quả.

***Bài 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.***

- Bước 1: nhập dữ liệu đầu vào là chuỗi str và tạo thêm 1 từ điển rỗng đi

- Bước 2:dùng for kết hợp if duyệt từng ký tự trong chuỗi và đếm số lần xuất hiện.

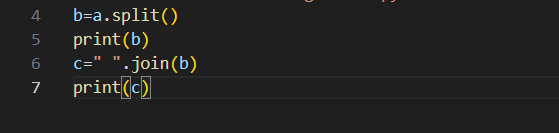


- Bước 3: chạy chương trình và in ra kết quả.

***Bài 10: viết chương trình sử dụng các phương thức split và join để tách nhập xâu ký tự.***

- Bước 1: khai báo chuỗi ta muốn nhập và tách.

- Bước 2: Tách chuỗi, in danh sách và ghép chuỗi lại thành danh sách.

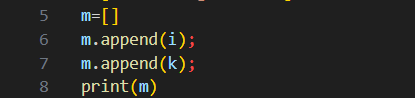


- Bước 3: chạy chương trình và in ra kết quả.

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

***-*** Bước 1: khởi tạo 2 danh sách I và k.

- Bước 2: khởi tạo danh sách rỗng m và thêm vào cho m.



- Bước 3: tạo 1 từ điển d.



- Bước 4: chạy chương trình và in ra kết quả.

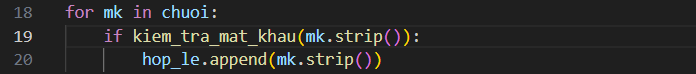
***Bài 12: một web yêu cầu người dùng nhập tên người dùng và mật khẩu để đăng ký. Viết chương trình để kiểm tra tính hợp lệ của mật khẩu mà người dùng nhập vào.***

- Bước 1: tạo hàm kiểm tra mật khẩu và các điều kiện kiểm tra trong hàm nếu thỏa mãn trả về True, nếu sai tra về False.

- Bước 2: nhập dữ liệu và tạo không gian rỗng hop\_le



- Bước 3: sử dụng for và if duyệt qua từng mật khẩu trong danh sách nếu hợp lệ thêm nó vào hop\_le.

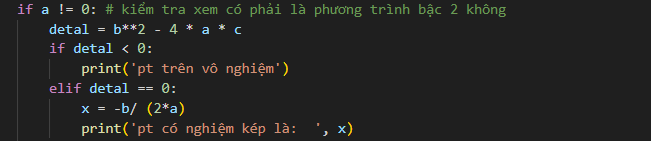


- Bước 4: chạy chương trình và in ra kết quả.

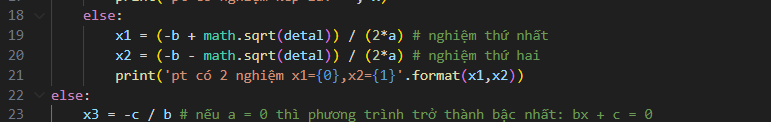
***Bài 13: Viết chương trình giải phương trình bậc 2: ax^2 + bx + c = 0, với các hệ số a,b,c nhập từ bàn phím***

- Bước 1: dùng import math để sử dụng hàm căn bậc 2 và nhập dữ liệu a,b,c từ bàn phím.

- Bước 2: dùng if, elif để kiểm tra các điều kiện của phương trình.



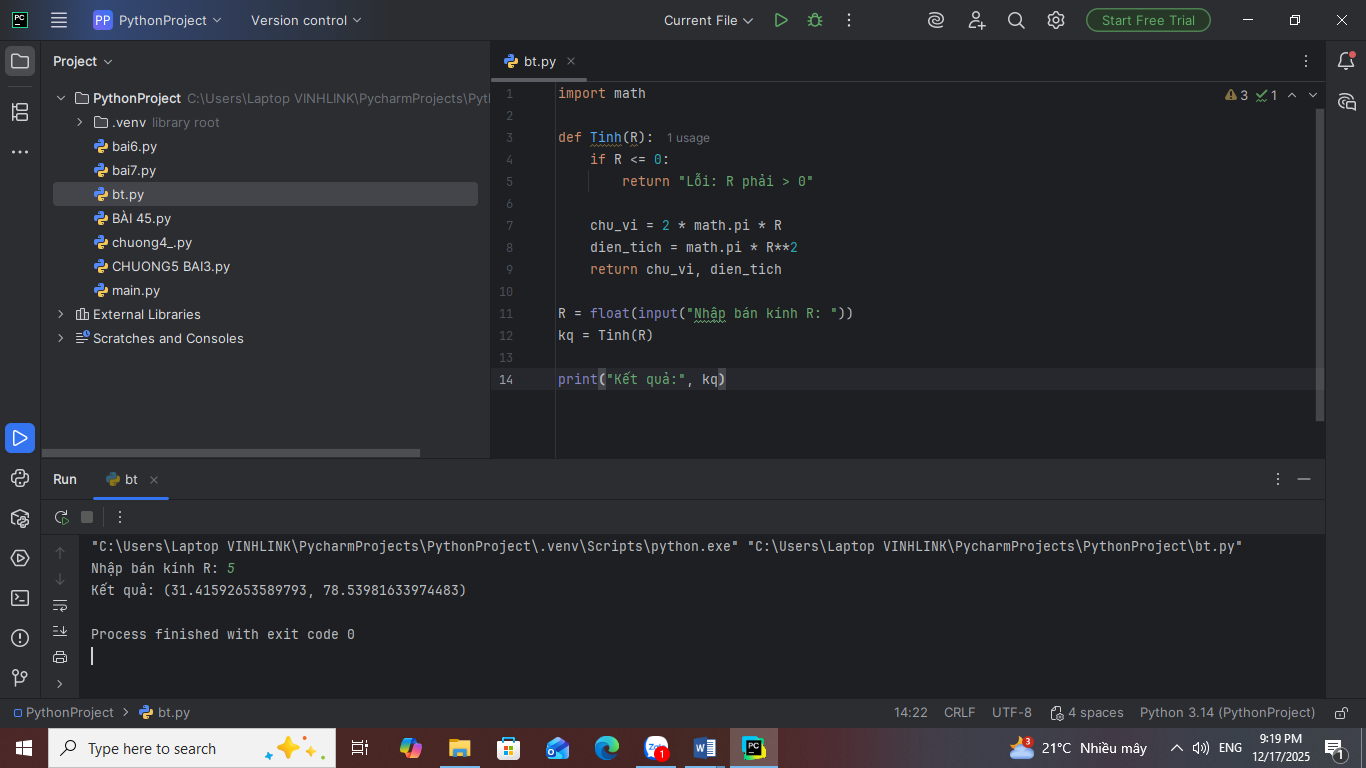
- Bước 3: tính ra nghiệm của phương trình và xét a = 0 thì đây là phương trình bậc nhất.



- Bước 4: khởi chạy chương trình và in ra kết quả.

**Chương 3: Lập trình hàm trong python.**

Câu 10.  
I.Mục đích  
 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.  
II.Các bước thực hiện và kết quả  
1.Nhập bán kính R  
 người dung nhập giá trị R từ bàn phím  
 Kết quả: có giá trị đầu vào để kiểm tra và tính toán  
2. Kiểm tra tính hợp lệ của R  
 Điều kiện hợp lệ: R > 0  
 Nếu R ≤ 0 → trả về thông báo lỗi.  
 Kết quả: đảm bảo chỉ tính toán với dữ liệu đúng.  
3. Tính chu vi  
 Công thức:C = 2 × π × R  
 Kết quả: nhận được giá trị chu vi chính xác.  
4. Tính diện tích  
 Công thức:S = π × R²  
 Kết quả: nhận được diện tích hình tròn.  
5. Trả kết quả  
 Hàm trả về bộ đôi: (chu vi, diện tích)  
 Kết quả: chương trình in ra giá trị đúng yêu cầu.



**Chương 4: Các kiểu dữ liệu có cấu trúc trong Python**

Câu 1,câu 2

I.Mục đích

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

Chỉnh sửa ví dụ trên: hãy bỏ qua không in ra những kí tự “không nhìn thấy” (dấu

space và dấu tab).

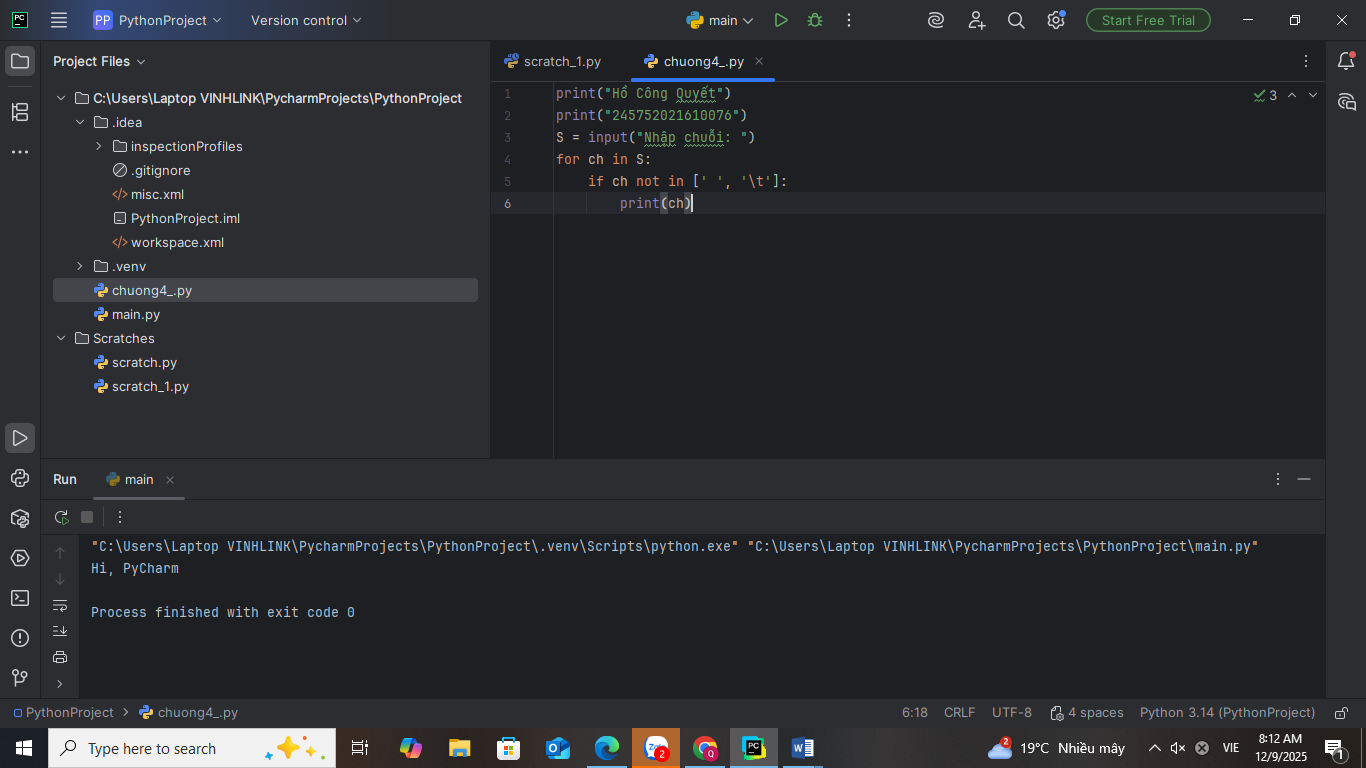
II.Các bước thực hiện và kết quả

1. Nhập chuỗi S.

2. Duyệt từng ký tự ch.

3. Kiểm tra: nếu ch không phải space (' ') và không phải tab ('\t') thì in ra; ngược lại bỏ qua.

Kết quả:Chỉ in các ký tự “nhìn thấy” (chữ, số, dấu câu,…), không in khoảng trắng và tab.



Câu 4,câu 5

I.Mục đích

-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

-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

II.Các bước thực hiện và kết quả

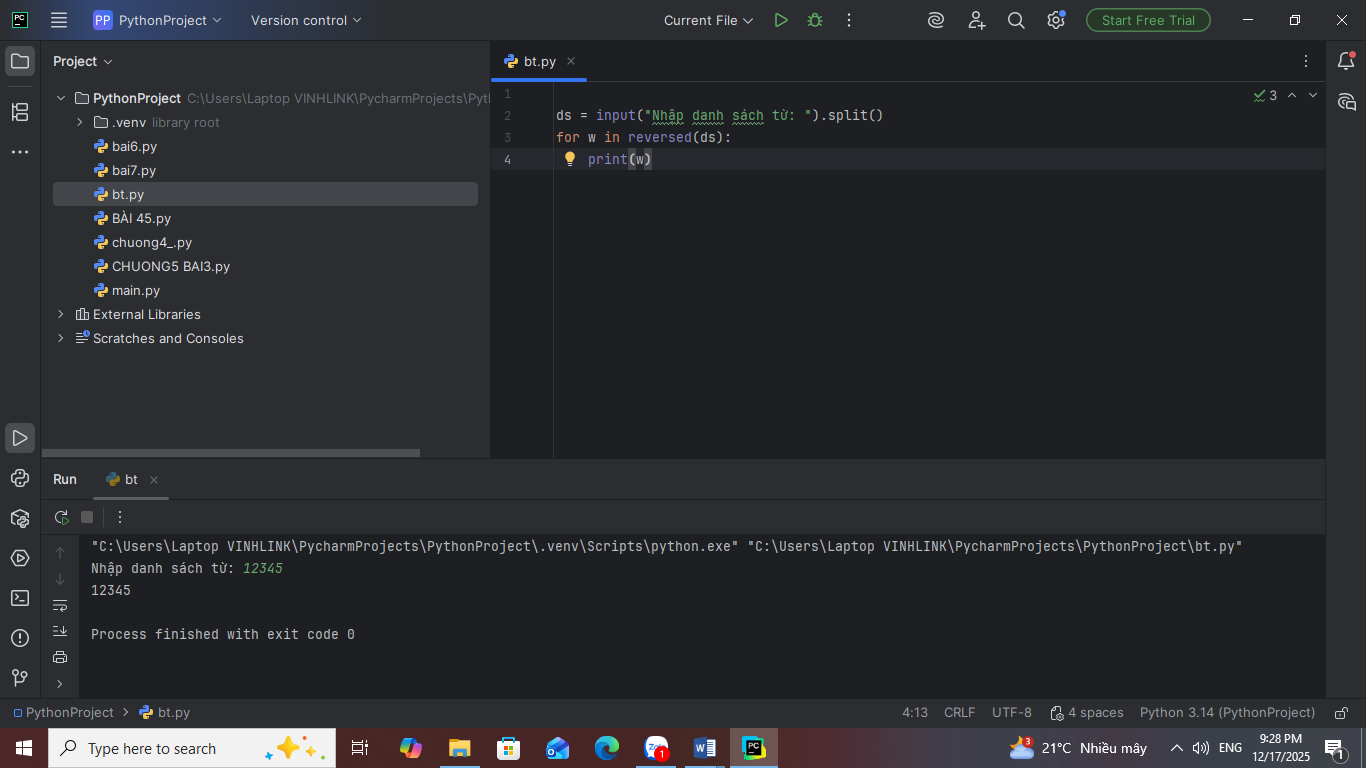
1. Dùng input() lấy chuỗi người dùng gõ.

2. Dùng split() (không truyền tham số) để tách chuỗi theo mọi whitespace (space, tab, newline).

3. print(ds) hiển thị danh sách dưới dạng ['a','b',...]. Nếu cần in từng phần tử, lặp qua ds và print(item).

Kết quả: ds là list các từ/ phần tử đã tách.

4. Nhập dòng chứa các từ.

6. Duyệt ds theo thứ tự đảo ngược (reversed(ds) hoặc ds[::-1]) và in từng từ.

Câu 6.

I.Mục đích

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).

II.Các bước thực hiện và kết quả

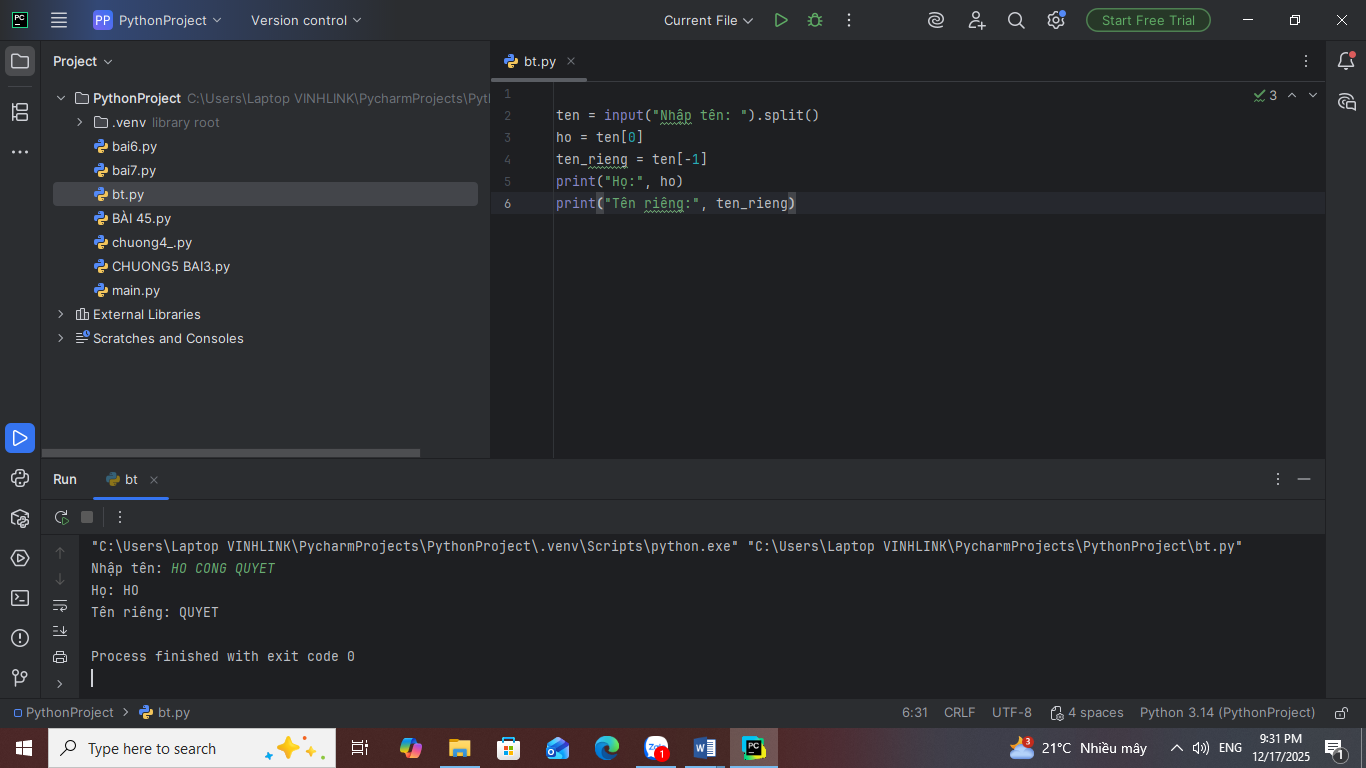
1. .strip() — loại khoảng trắng đầu/cuối để tránh lỗi khi người dùng bấm space trong đầu/cuối.

2. split() — tách chuỗi theo mọi whitespace (space, tab, nhiều khoảng trắng) thành danh sách từ parts.

3. Lấy ho = parts[0] (phần tử đầu) và ten\_rieng = parts[-1] (phần tử cuối)

4. In ra họ và tên riêng.

Kết quả: Với giả thiết họ và tên riêng mỗi phần chỉ gồm một âm, cách trên đúng: họ lấy phần đầu, tên riêng lấy phần cuối.



Câu 7.

I.Mục đích

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

II.Các bước thực hiện và kết quả

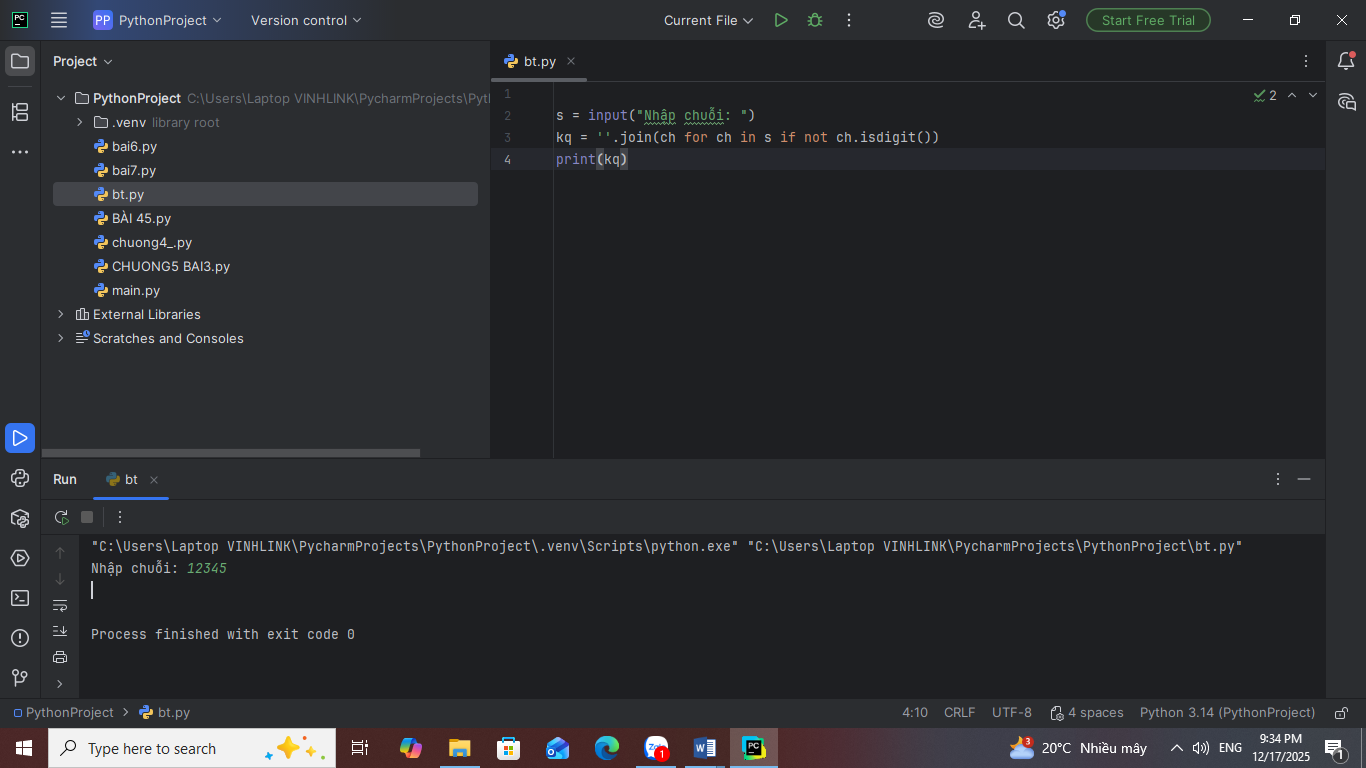
1. Đọc chuỗi từ bàn phím bằng input().

2. Duyệt từng ký tự trong chuỗi; với mỗi ký tự kiểm tra ch.isdigit() (True nếu ký tự là chữ số 0–9).

3. Giữ lại những ký tự không phải chữ số — ghép lại bằng ''.join(...).

4. In chuỗi kết quả.

Kết quả: Kết quả là một chuỗi mới giống chuỗi gốc nhưng đã loại bỏ mọi chữ số (0–9).



Câu 8.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. strip() — loại khoảng trắng đầu/cuối để tránh chuỗi rỗng vô ý.

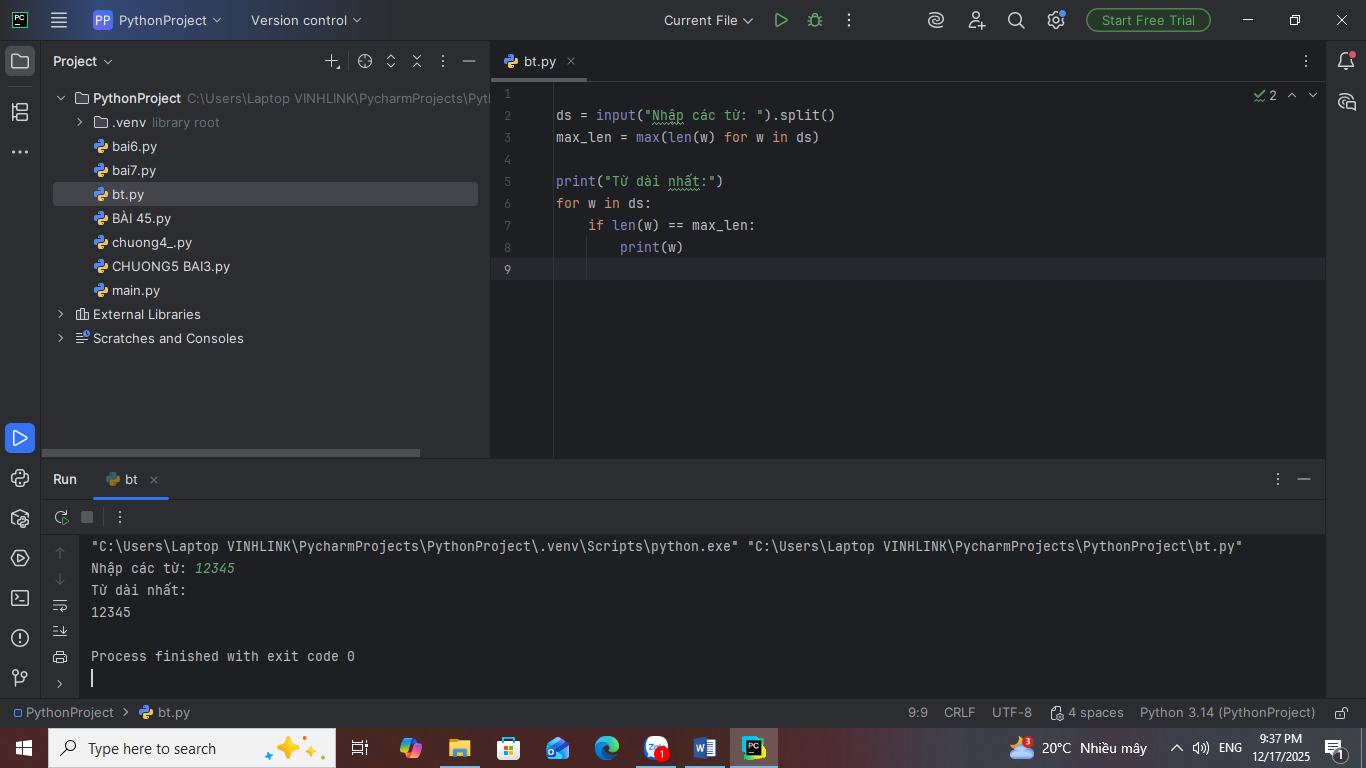
2. split() — tách chuỗi đầu vào thành danh sách words theo mọi whitespace (space, tab, nhiều space đều được xử lý).

3. Tính max\_len = độ dài lớn nhất bằng max(len(w) for w in words).

4. Lấy danh sách longest gồm mọi từ có len(w) == max\_len (giữ thứ tự xuất hiện ban đầu).

5. In max\_len và in từng từ trong longest.

Kết quả: in độ dài lớn nhất và tất cả từ có độ dài đó, mỗi từ trên một dòng, theo thứ tự xuất hiện trong input.



Câu 15.

I.Mục đích

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

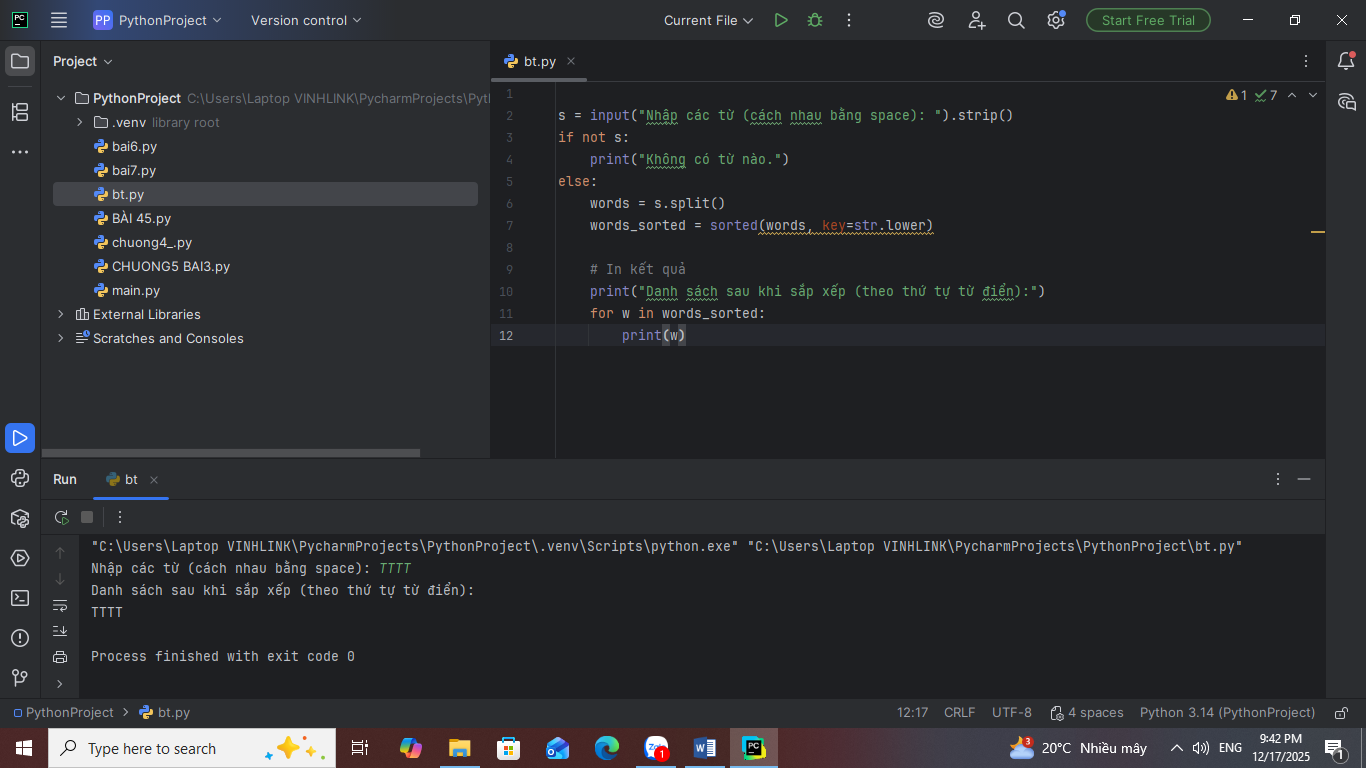
II.Các bước thực hiện và kết quả

1. input().strip() — đọc dòng vào và loại khoảng trắng đầu/cuối.

2. split() — tách chuỗi thành danh sách từ theo mọi whitespace.

3. sorted(..., key=str.lower) — sắp xếp danh sách theo thứ tự từ điển không phân biệt hoa/thường

4. In danh sách đã sắp xếp, mỗi từ trên một dòng (đọc dễ).



Câu 16.

I.Mục đích

.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

II.Các bước thực hiện và kết quả

1. Nhập chuỗi

Dùng input().strip() để nhận toàn bộ chuỗi do người dùng nhập.

.strip() giúp loại bỏ khoảng trắng đầu/cuối tránh lỗi.

2. Tách chuỗi theo dấu phẩy

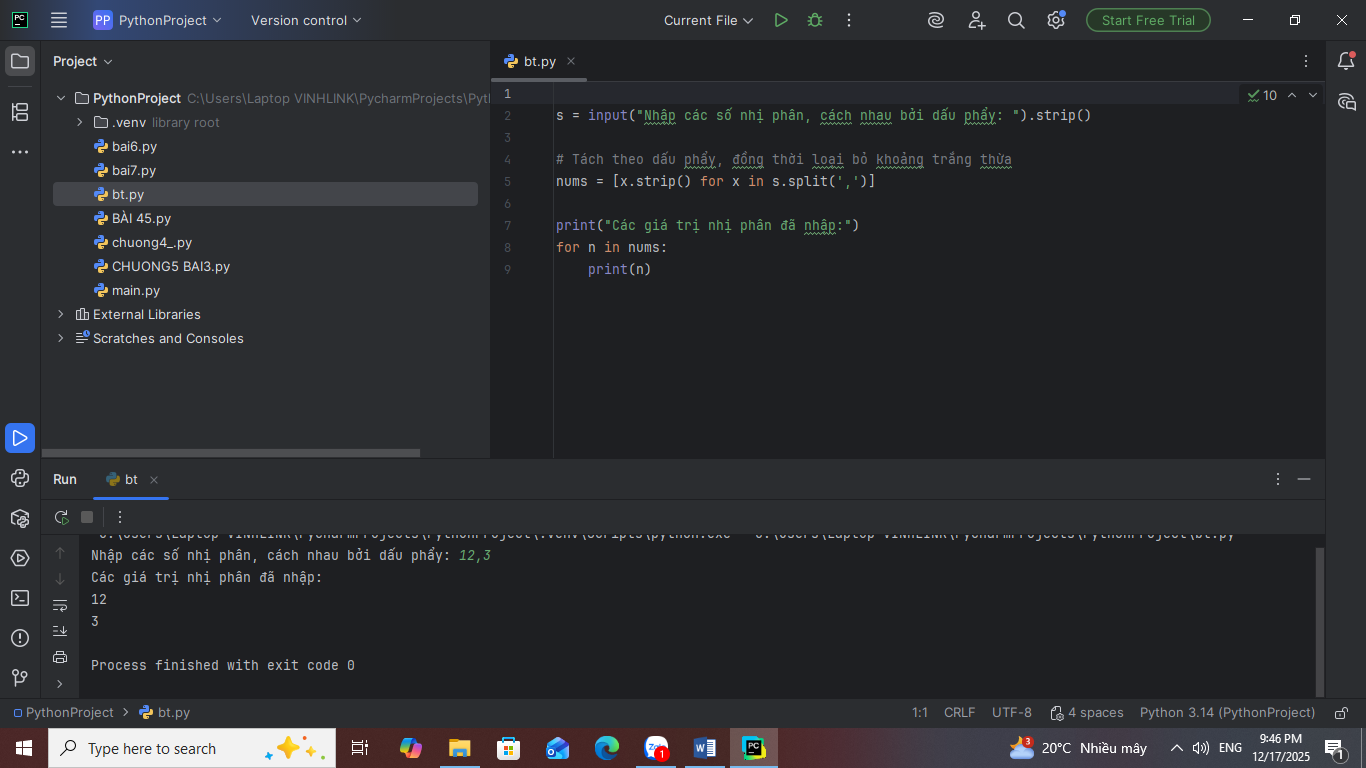
Dùng s.split(',') để tách thành danh sách các số nhị phân.

Dùng x.strip() để loại bỏ khoảng trắng thừa sau dấu phẩy

3. In từng giá trị

Duyệt danh sách và in từng số nhị phân lên màn hình.

Giữ nguyên thứ tự như lúc người dùng nhập.

Kết quả: Chương trình hiển thị lại toàn bộ các chuỗi nhị phân mà người dùng đã nhập, đúng theo thứ tự ban đầu.

Câu 17.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. Đọc n: chuyển chuỗi nhập thành int. Nếu không phải số nguyên thì báo lỗi.

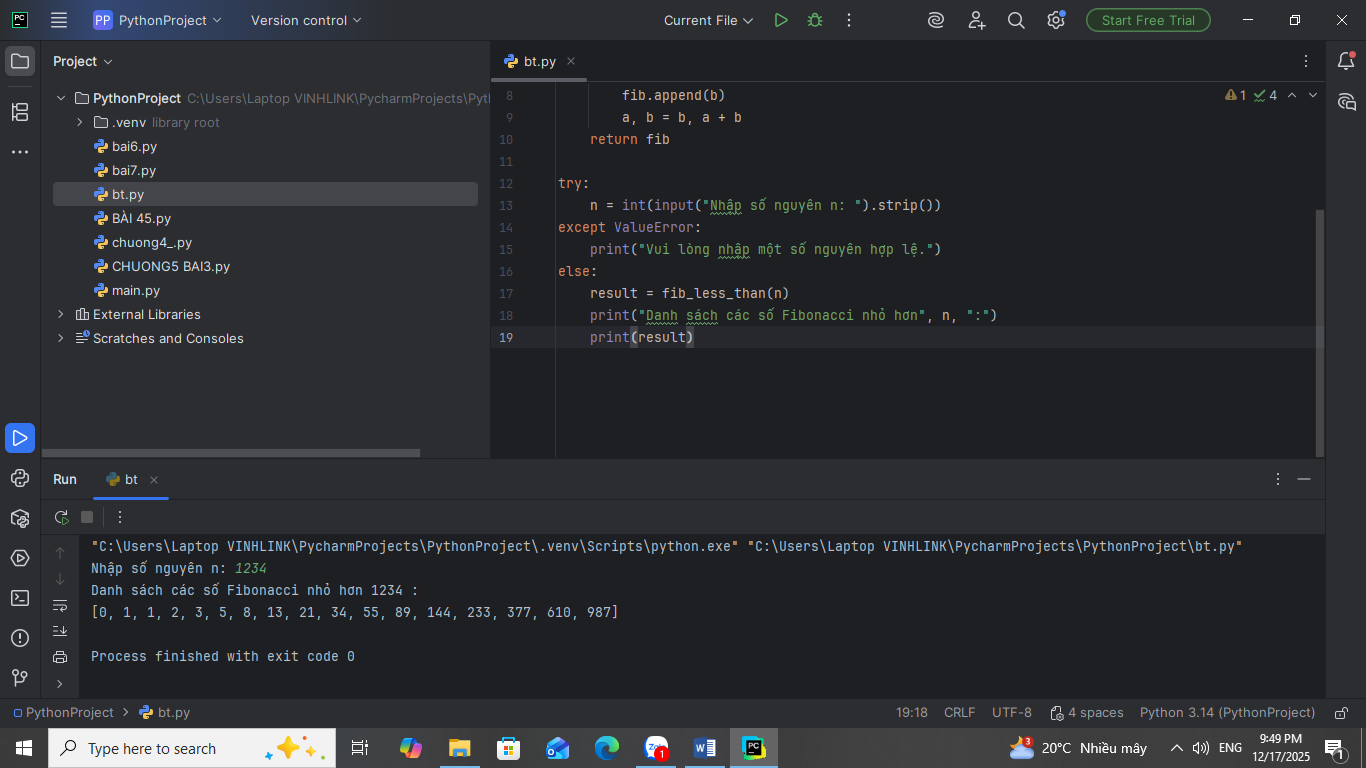
2. Xử lý trường hợp đặc biệt: nếu n <= 0 trả về danh sách rỗng [].

3. Khởi tạo dãy Fibonacci: bắt đầu với a=0, b=1 và danh sách fib = [0] (chứa 0 nếu 0 < n).

4. Sinh dãy: lặp while b < n: thêm b vào danh sách rồi cập nhật (a,b) <- (b, a+b). Dừng khi tiếp giá trị tiếp theo >= n

5. In kết quả: in danh sách Fibonacci thu được (theo thứ tự tăng dần, tất cả các phần tử đều < n).

Kết quả: Kết quả là một list chứa tất cả số Fibonacci nhỏ hơn n, bắt đầu từ 0 (nếu 0 < n).



Câu 19

I.Mục đích

dùng python Hãy tạo ra tuple P gồm các số nguyên tố nhỏ hơn 1 triệu

II.Các bước thực hiện và kết quả

1. Chọn thuật toán tìm số nguyên tố

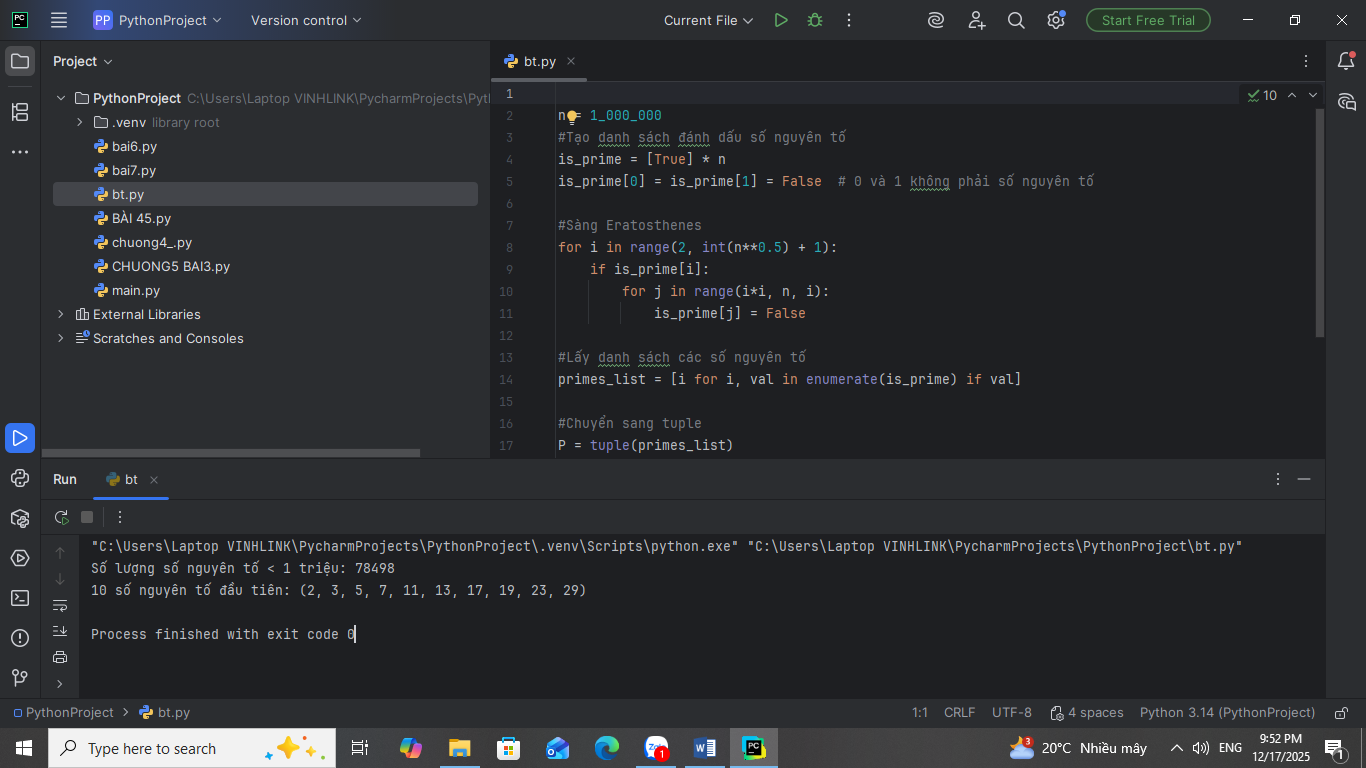
Tạo một danh sách True với các chỉ số từ 0 đến 999,999.

Đánh dấu các bội số của từng số nguyên tố là False.

Chỉ các số còn True là số nguyên tố.

2. tuple(primes): lưu trữ các số nguyên tố dưới dạng tuple P (immutable).

Kết quả:Số lượng số nguyên tố < 1 triệu: 78,498 số.



Câu 20.

I.Mục đích

Nhập n, in n dòng đầu tiên của tam giác pascal

II.Các bước thực hiện và kết quả

1. Nhập dữ liệu

Nhập số nguyên n từ người dùng, xác định số dòng cần in.

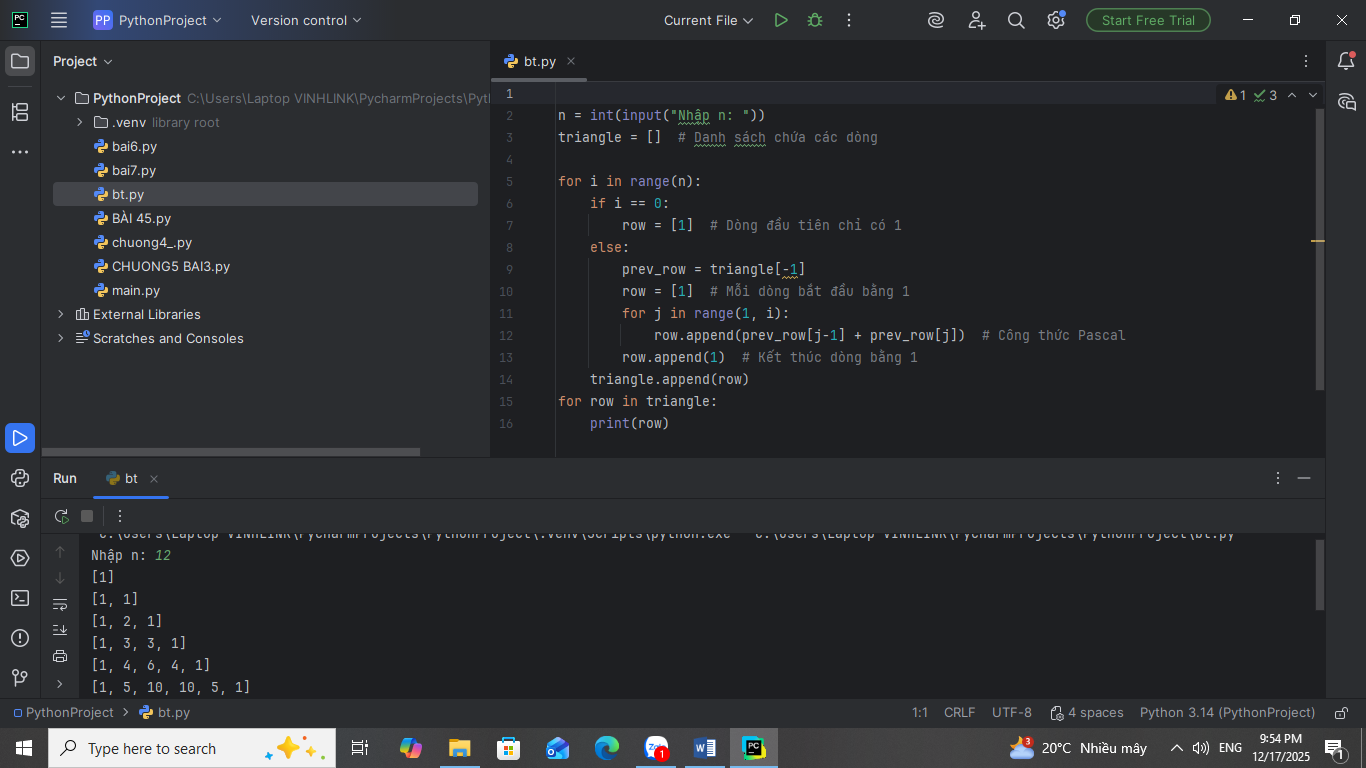
2. Tạo tam giác Pascal

Dòng đầu tiên luôn [1].

Dòng thứ i được tính từ tổng hai số liền kề ở dòng i-1.

Mỗi dòng bắt đầu và kết thúc bằng 1.

3. In tam giác Pascal



Câu 21.

I.Mục đích

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

II.Các bước thực hiện và kết quả

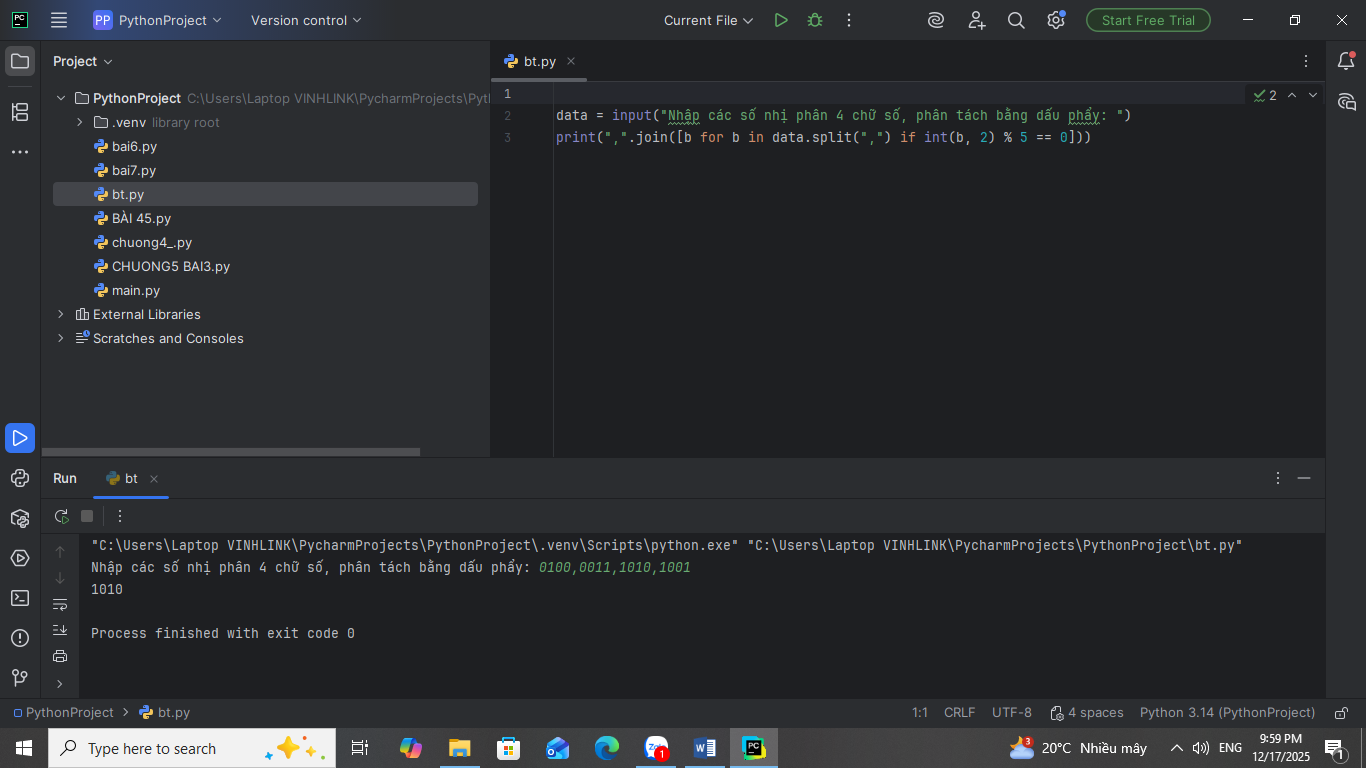
1. Nhập chuỗi nhị phân, tách thành danh sách.

2. Chuyển từng số nhị phân sang thập phân.

3. Kiểm tra chia hết cho 5.

4. Lưu các số chia hết và in ra, phân tách bằng dấu phẩy.

Kết quả: chỉ in ra các số nhị phân chia hết cho 5



Câu 22.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. Duyệt các số trong khoảng 1000 đến 3000

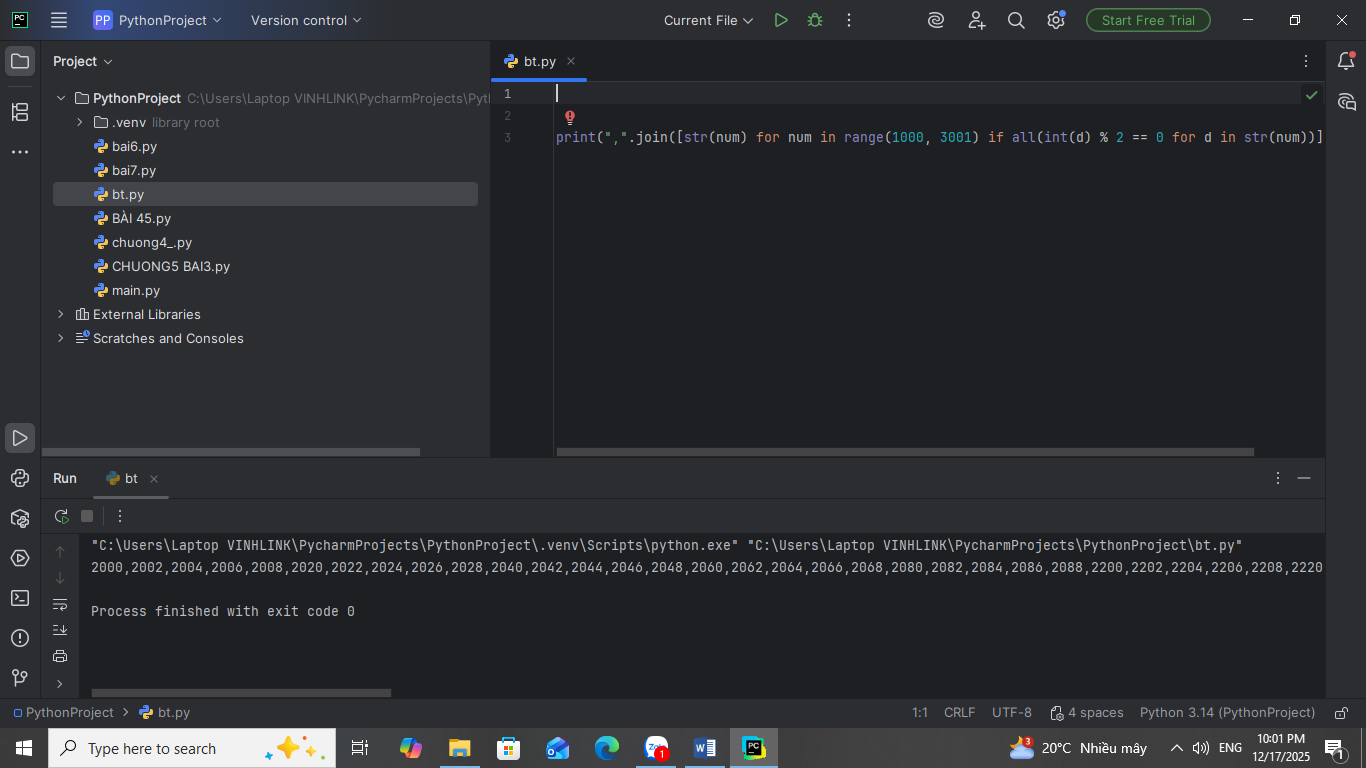
range(1000, 3001) duyệt từ 1000 đến 3000 (bao gồm 3000).

2. str(num) → chuyển số thành chuỗi để kiểm tra từng chữ số.

3. all(int(d) % 2 == 0 for d in str(num)) → lọc các số mà tất cả chữ số đều chẵn.

4. ",".join([...]) → nối các số tìm được thành chuỗi, cách nhau bằng dấu phẩy, in ra một dòng.

Kết quả: In ra các chữ số chẵn thành chuỗi cách nhau bởi dấu phẩy.



Câu 23.

I.Mục đích

Viết một chương trình chấp nhận đầu vào là một câu, đếm số chữ cái và chữ số trong câu đó.

II.Các bước thực hiện và kết quả

1. Nhập câu từ người dùng

input() nhận một chuỗi từ người dùng.

2. Khởi tạo biến đếm

letters → đếm số chữ cái (a-z, A-Z).

digits → đếm số chữ số (0-9).

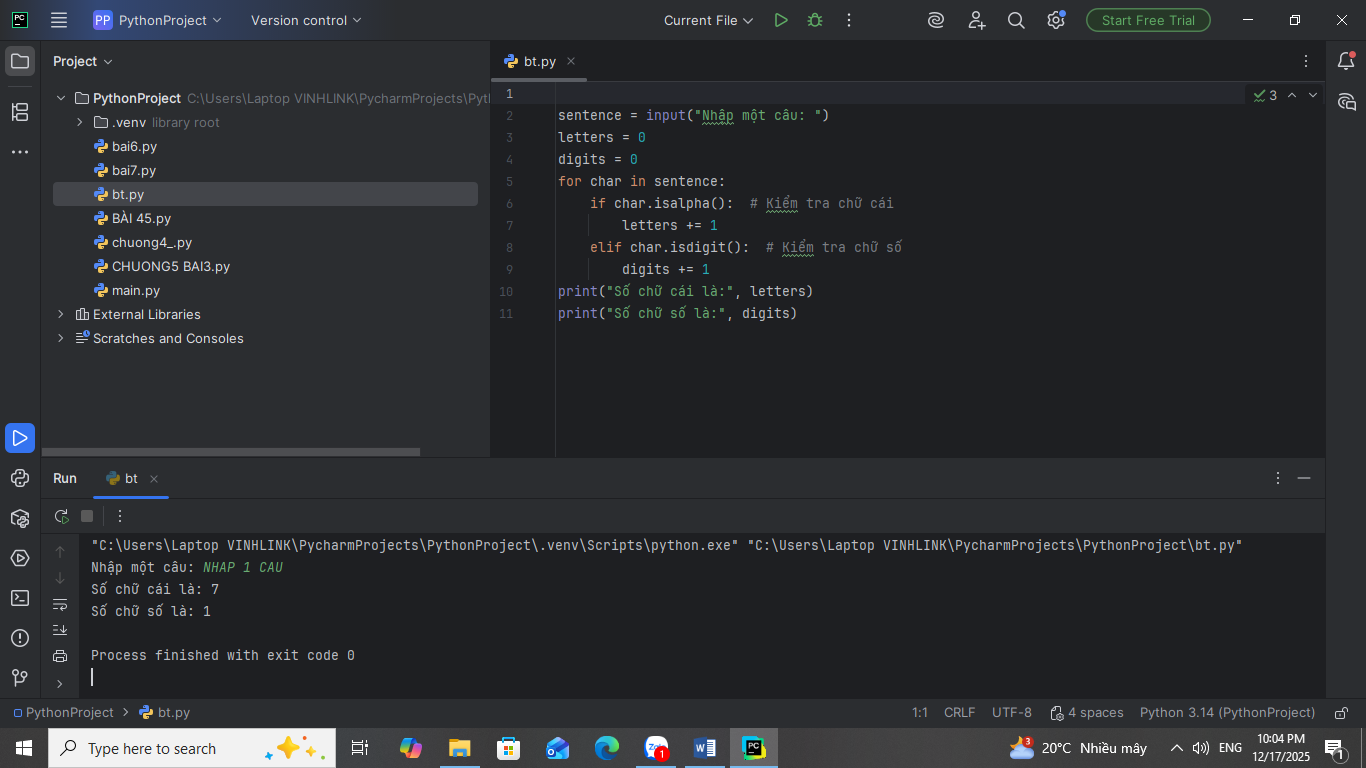
3. Duyệt từng ký tự và kiểm tra

char.isalpha() → True nếu ký tự là chữ cái.

char.isdigit() → True nếu ký tự là số.

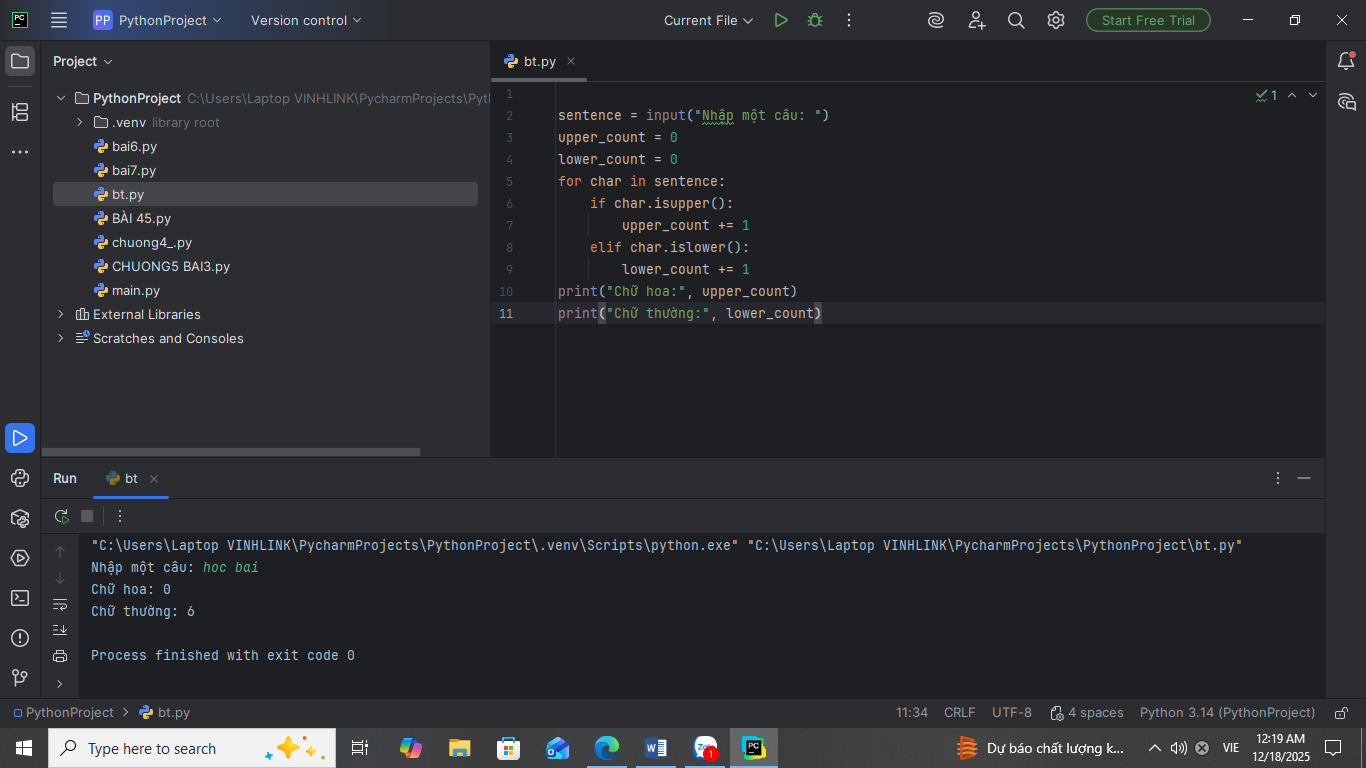
Cộng dồn vào biến tương ứng.

Kết quả: In ra màn hình số chữ cái và số chữ số.

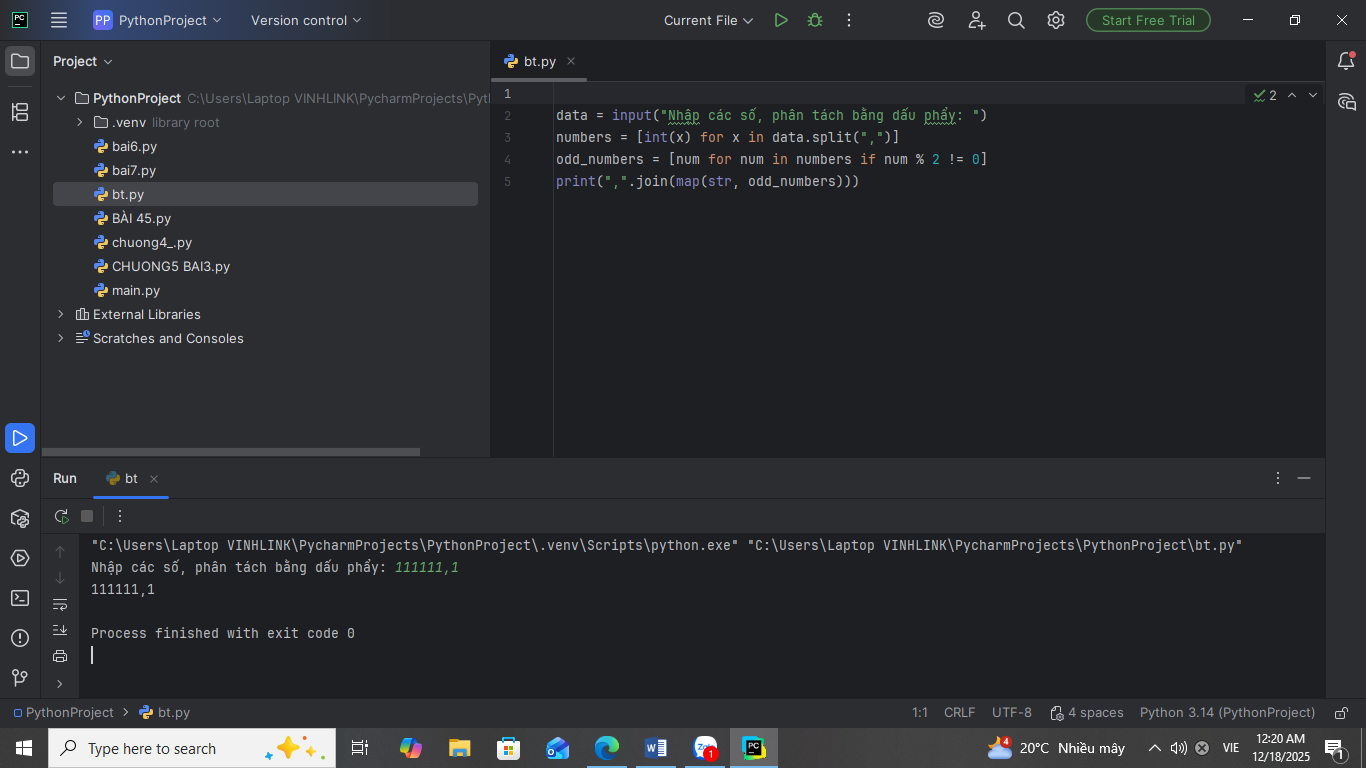


Câu 24.

I.Mục đích  
 Viết một chương trình chấp nhận đầu vào là một câu, đếm chữ hoa, chữ thường.  
II.Các bước thực hiện và kết quả  
1. Nhập câu  
 Nhận chuỗi từ người dùng.  
2. Khởi tạo biến đếmCâu 24  
 upper\_count → đếm chữ hoa (A-Z).  
 lower\_count → đếm chữ thường (a-z).  
3. Duyệt từng ký tự và kiểm tra  
 char.isupper() → True nếu là chữ hoa.  
 char.islower() → True nếu là chữ thường.  
Kết quả: in ra màn hình số chữ hoa và số chữ thường.



Câu 25.  
I.Mục đích  
 Sử dụng một danh sách để lọc các số lẻ từ danh sách được người dùng nhập vào  
II.Các bước thực hiện và kết quả  
1. Nhập danh sách từ người dùng  
 input() nhận chuỗi  
 split(",") → tách chuỗi thành danh sách các chuỗi số.  
 int(x) → chuyển chuỗi thành số nguyên.  
 2. Lọc các số lẻ  
 Dùng list comprehension.  
 num % 2 != 0 → kiểm tra số lẻ.  
 Lưu các số lẻ vào danh sách odd\_numbers.  
3. In kết quả  
 map(str, odd\_numbers) → chuyển các số thành chuỗi.  
 ",".join(...) → nối các số bằng dấu phẩy.  
 Kết quả: in ra màn hình các số lẻ cách nhau bởi dấu phẩy.



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

Câu 3.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. Import thư viện NumPy

Dùng import numpy as np.

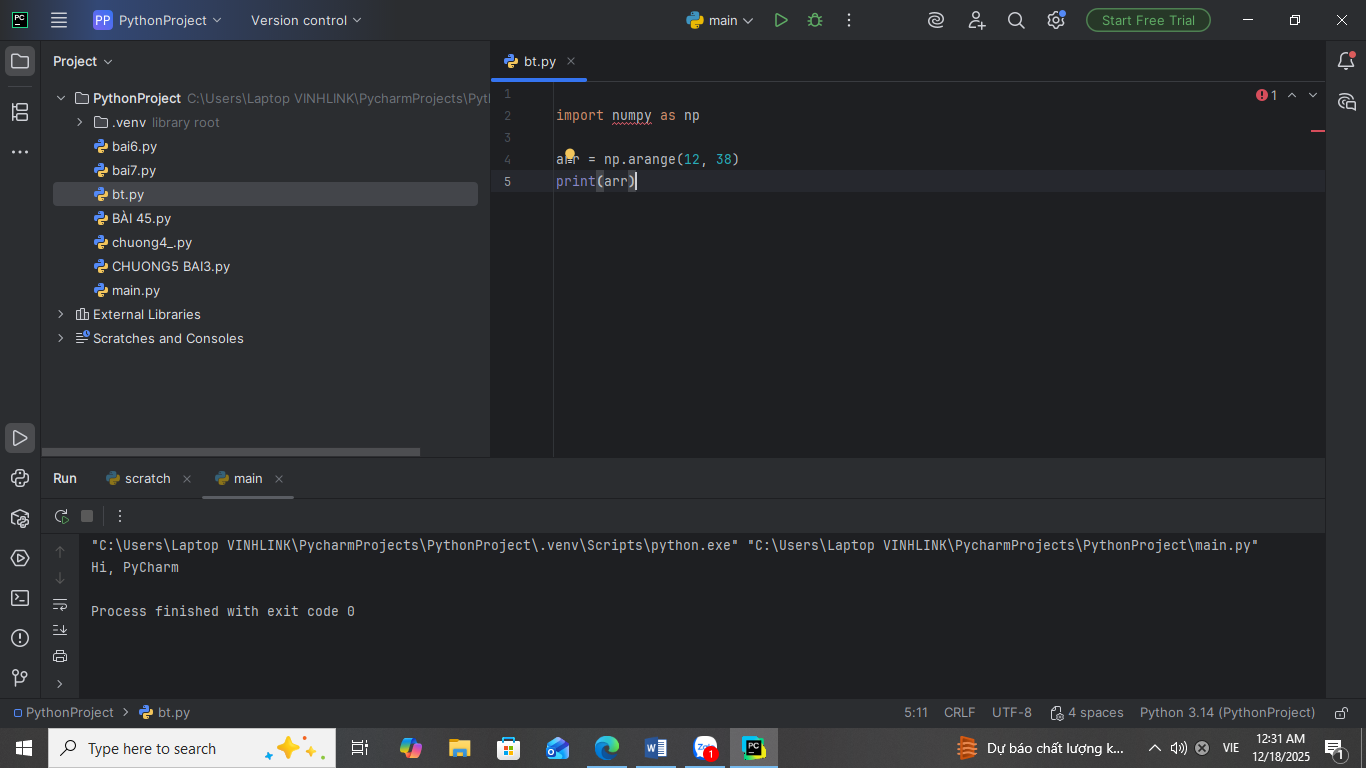
Kết quả: Sẵn sàng sử dụng hàm tạo mảng của NumPy.

2. Tạo mảng bằng np.arange(12, 38)

arange(start, stop) tạo dãy số từ start đến stop - 1.

Vì vậy arange(12, 38) → từ 12 đến 37.

Kết quả: Thu được mảng số nguyên liên tiếp.

3. In mảng kết quả

Câu 4.

I.Mục đích

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)

II.Các bước thực hiện và kết quả

1. Import thư viện NumPy

import numpy as np

Kết quả: Có thể dùng hàm tạo mảng.

2. Tạo mảng từ 12 đến 37

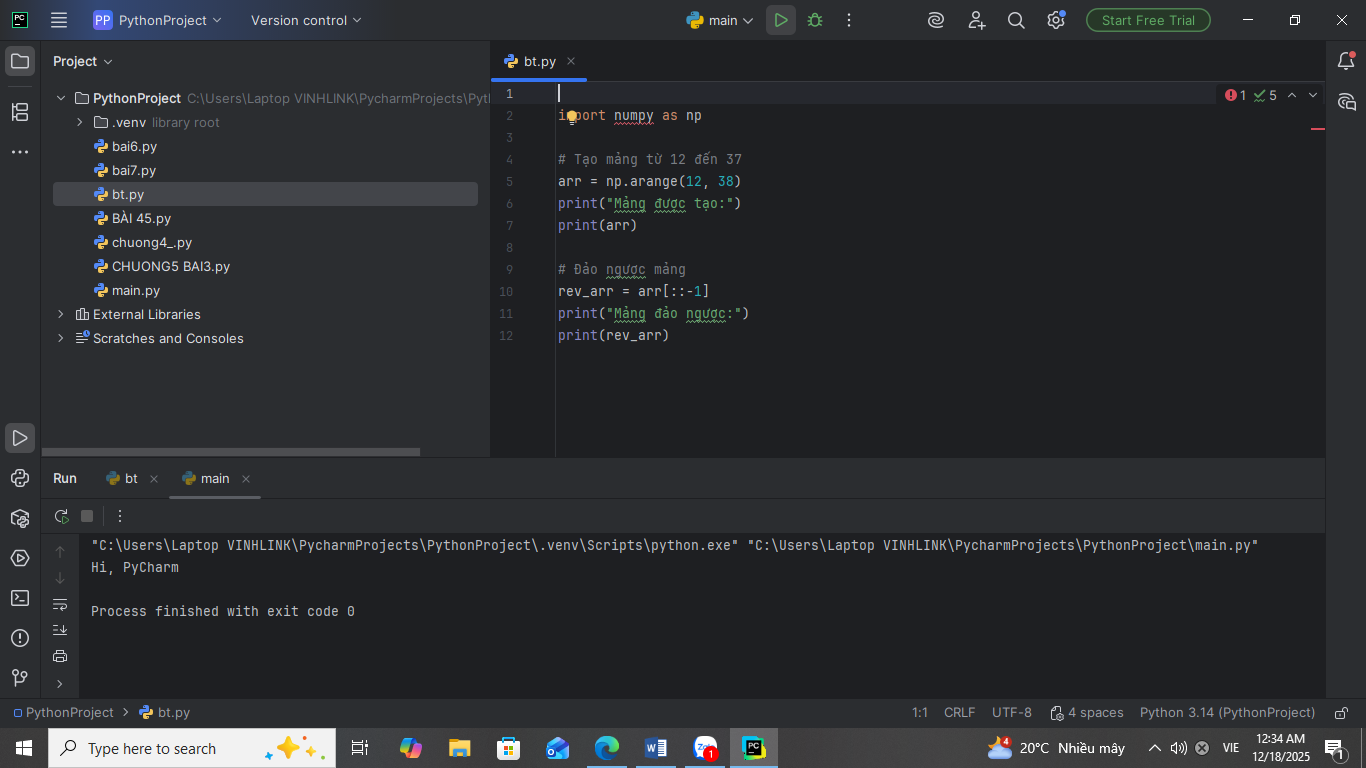
Dùng np.arange(12, 38)

Kết quả: Thu được mảng giá trị liên tiếp từ 12 đến 37.

3. Đảo ngược mảng

Dùng cú pháp cắt mảng arr[::-1]

Kết quả: Mảng bị đảo ngược hoàn toàn, phần tử đầu thành cuối.

4. In kết quả

Câu 11

I.Mục đích

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 )]

II.Các bước thực hiện và kết quả

1. Khai báo kiểu dữ liệu cấu trúc

Dùng dt = [('name','U10'), ('class','i4'), ('height','f4')].

Kết quả: Mỗi phần tử gồm 3 trường: tên (chuỗi), lớp (số nguyên), chiều cao (số thực).

2. Nhập dữ liệu danh sách sinh viên

Tạo danh sách tuple: tên, lớp, chiều cao.

Kết quả: Có dữ liệu ban đầu để chuyển vào NumPy.

3. Tạo mảng cấu trúc

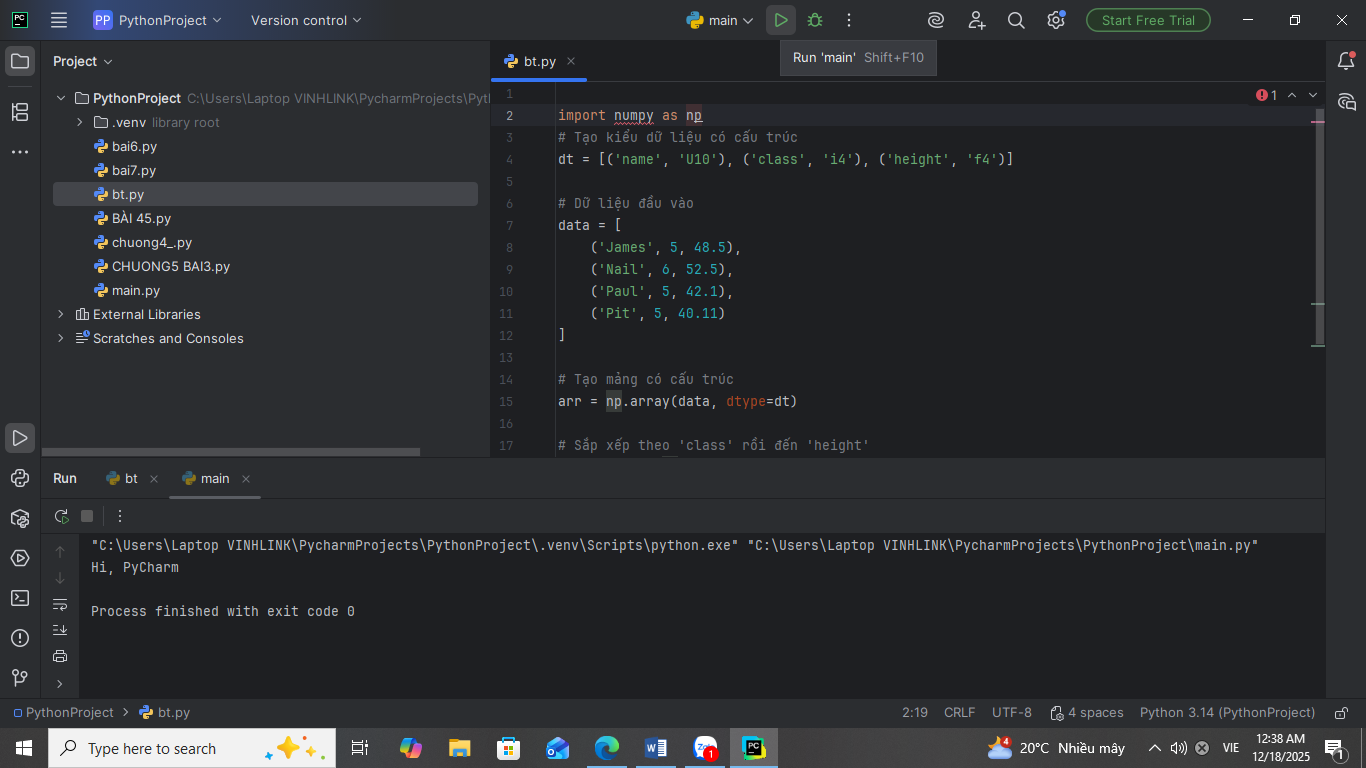
arr = np.array(data, dtype=dt)

Kết quả: Mảng NumPy có thể truy cập theo từng trường như một bảng.

4. Sắp xếp theo class, sau đó height

Dùng np.sort(arr, order=['class', 'height']).

Kết quả:Các sinh viên lớp 5 được sắp xếp theo chiều cao tăng dầnLớp 6 đứng sau.



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

Câu 2.

I.Mục đích

Định nghĩa class có tên là Hinhchunhat được xây dựng bằng chiều dài và chiều

rộng. Class Hinhchunhat có method để tính diện tích.

IICác bước thực hiện và kết quả

1. Tạo class Hinhchunhat

Định nghĩa class Hinhchunhat: trong Python.

Kết quả: tạo ra một kiểu dữ liệu mới mô tả hình chữ nhật.

2. Viết phương thức khởi tạo \_\_init\_\_

Nhận vào hai tham số: dai và rong.

Gán vào thuộc tính self.dai, self.rong.

Kết quả: mỗi đối tượng sẽ có chiều dài và chiều rộng riêng.

3. Viết method tính diện tích

Tạo hàm dientich(self) trả về dai \* rong.

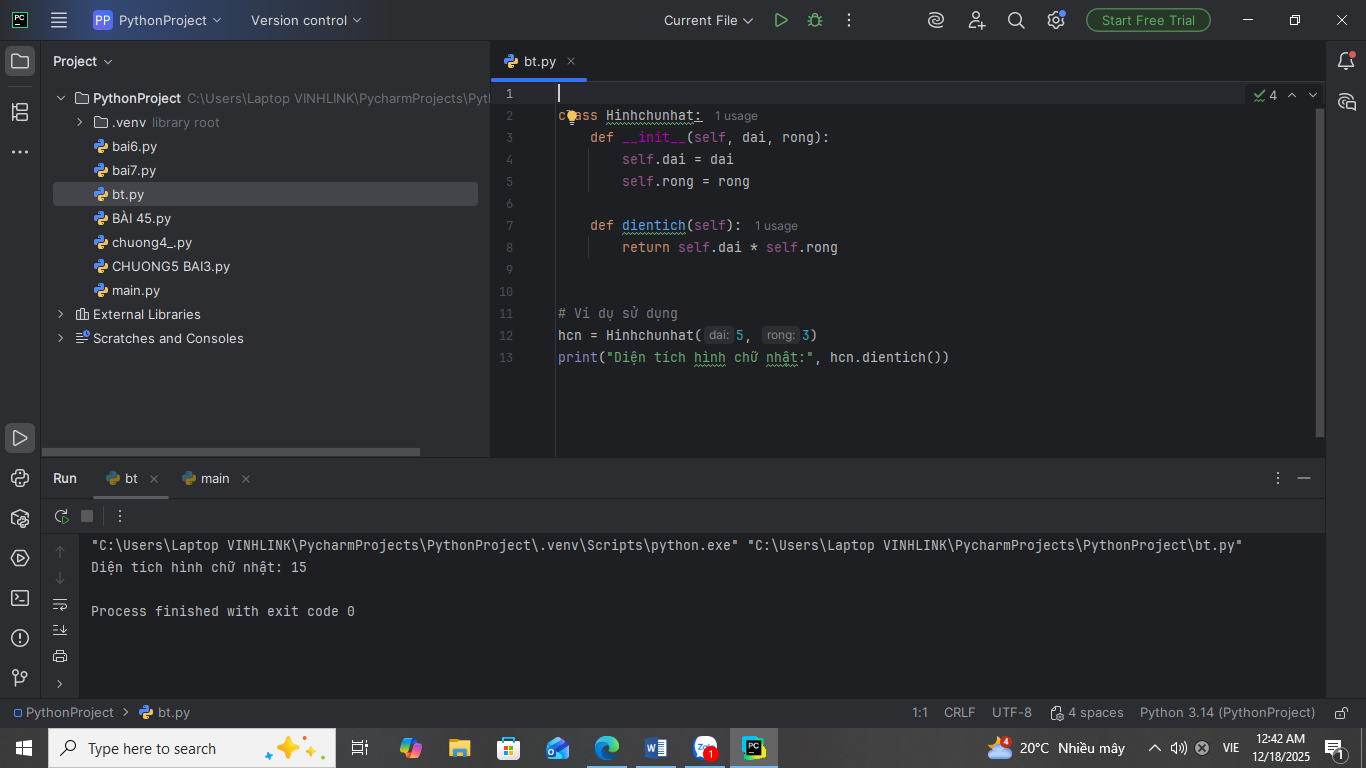
Kết quả: đối tượng có thể tự tính diện tích.

4. Tạo đối tượng và gọi phương thức

Ví dụ: hcn = Hinhchunhat(5, 3).

Gọi hcn.dientich().

Kết quả: hiển thị diện tích: 15



Câu 4.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. Tạo class RomanConverter

Định nghĩa class RomanConverter:.

Kết quả: tạo lớp xử lý chuyển đổi ký số La Mã.

2. Khởi tạo bảng giá trị La Mã

Trong \_\_init\_\_, tạo dictionary chứa giá trị của từng ký tự: I, V, X,…

Kết quả: chương trình có dữ liệu để tra cứu.

3. Xây dựng phương thức roman\_to\_int

Duyệt từng ký tự của chuỗi.

Nếu ký tự hiện tại nhỏ hơn ký tự sau → trừ.

Ngược lại → cộng.

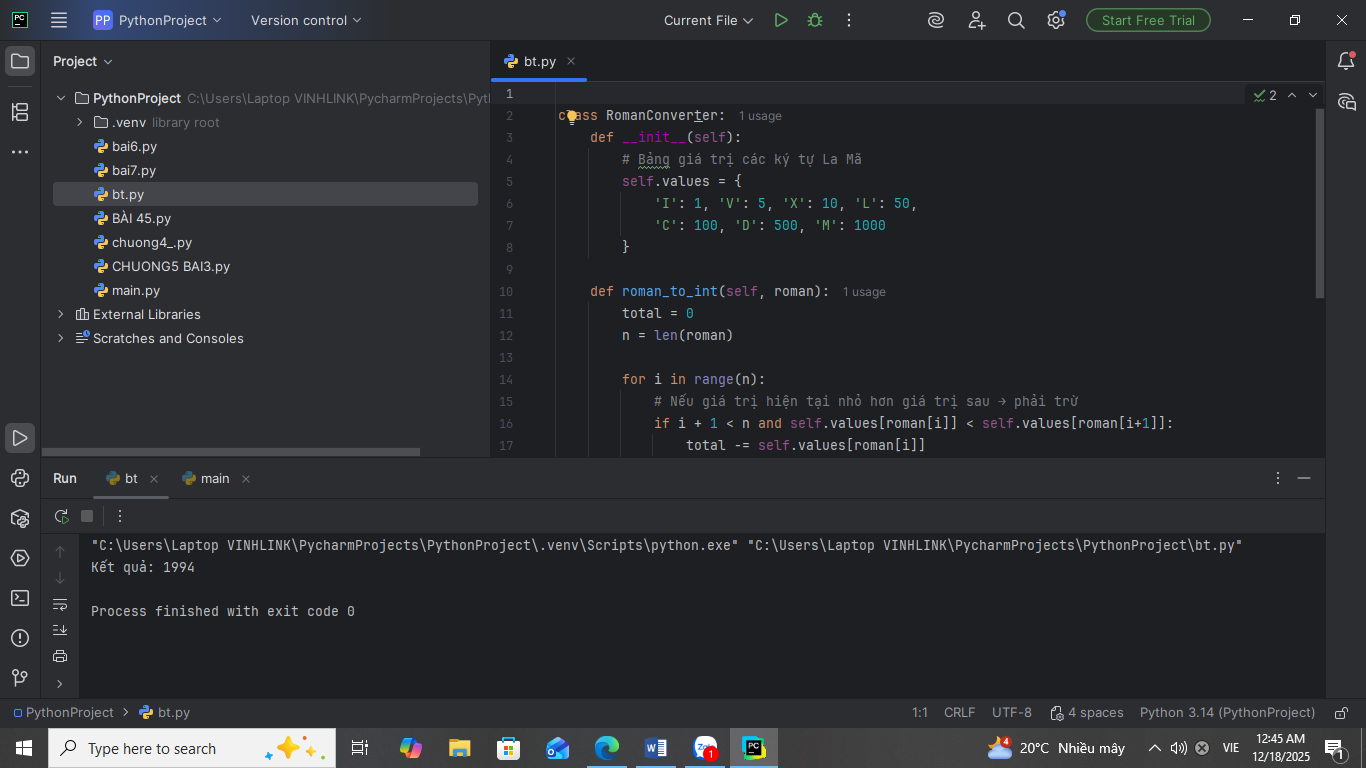
Kết quả: phép chuyển đổi đúng theo quy tắc số La Mã.

4. Tạo đối tượng và gọi phương thức

converter = RomanConverter()

Gọi converter.roman\_to\_int("MCMXCIV").

Kết quả: chương trình trả về số nguyên tương ứng, ví dụ:1994.



Câu 5.

I.Mục đích

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

Dữ liệu vào : 'hello .py'

Đầu ra : '.py hello'

II.Các bước thực hiện và kết quả

1. Tạo class ReverseWords

Class nhận vào chuỗi ban đầu.

Kết quả: lưu được chuỗi cần xử lý.

2. Viết method reverse()

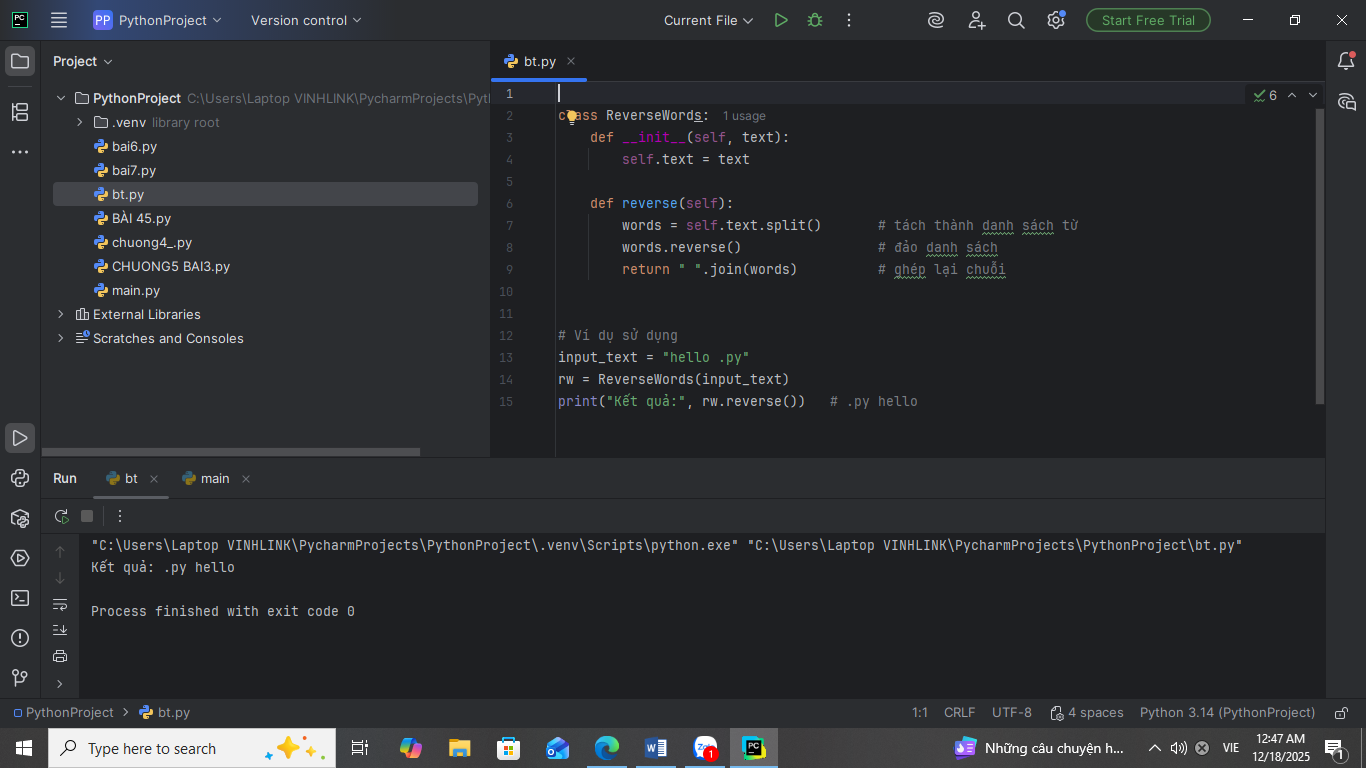
split() → tách chuỗi thành các từ: ["hello", ".py"]

reverse() → đảo thứ tự danh sách: [".py", "hello"]

join() → ghép lại thành chuỗi mới.

Kết quả: tạo ra chuỗi đảo ngược theo từng từ.

3. Tạo đối tượng và chạy



Câu 6.

I.Mục đích

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

II.Tóm tắt các bước thực hiện và kết quả

1. Khởi tạo class

Định nghĩa class MyString với thuộc tính self.s rỗng.

Kết quả: Có đối tượng chứa nơi lưu chuỗi.

2. Lấy chuỗi từ người dùng (get\_String)

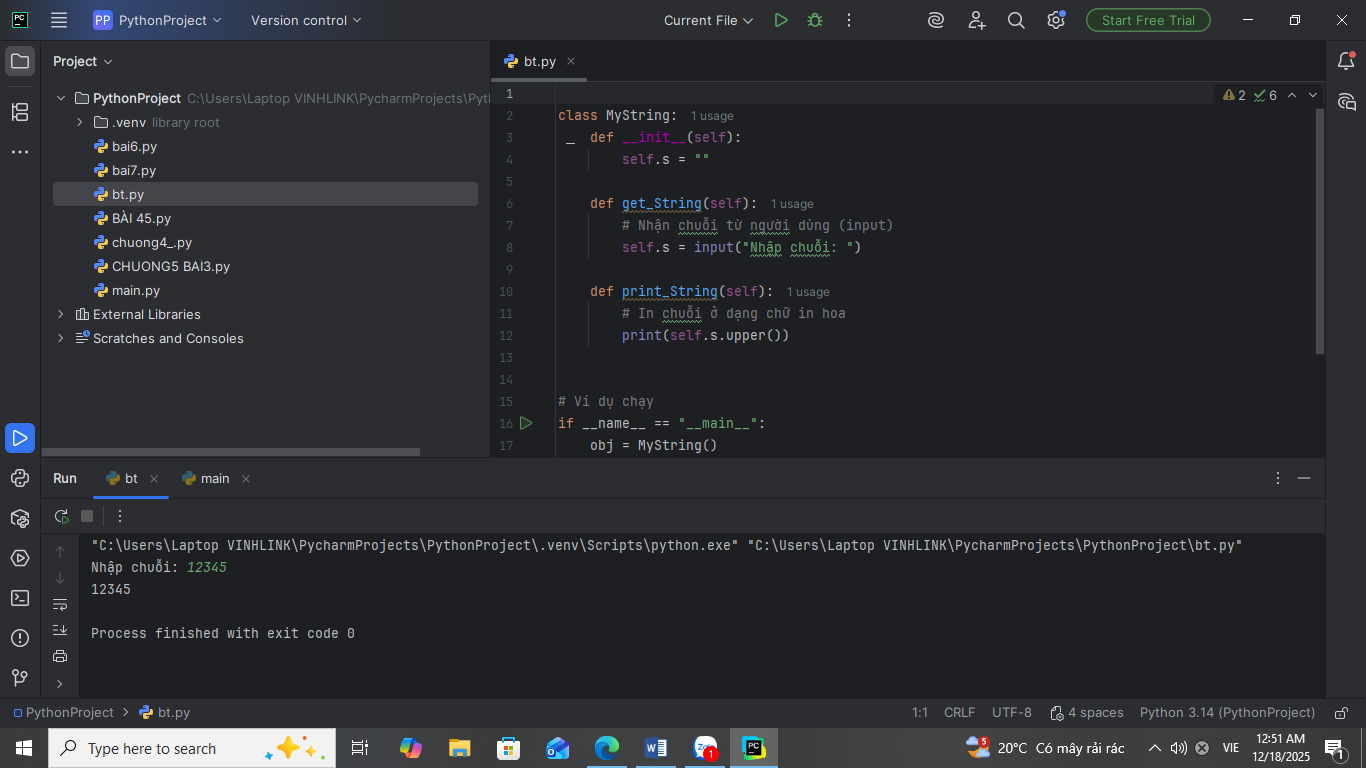
Gọi input() và gán giá trị vào self.s.

Kết quả: self.s chứa chuỗi người dùng nhập (ví dụ “đại học vinh").

3. In chuỗi ở dạng chữ in hoa (print\_String)

Dùng self.s.upper() và print() để xuất.

Kết quả: Màn hình hiển thị chuỗi in hoa (ví dụ: ĐẠI HỌC VINH").



Câu 7.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. Định nghĩa class Circle

Viết class Circle: chứa phương thức khởi tạo \_\_init\_\_.

Kết quả: Tạo kiểu dữ liệu đại diện cho một hình tròn.

2. Khởi tạo với bán kính

\_\_init\_\_(self, radius) lưu radius vào self.radius và kiểm tra radius >= 0.

Kết quả: Đối tượng Circle có thuộc tính bán kính hợp lệ.

3. Viết phương thức area()

Công thức: area = π \* r^2 (dùng math.pi).

Kết quả: Trả về diện tích dạng số thực. (Với r=3 → ≈ 28.2743338823)

4. Viết phương thức circumference()

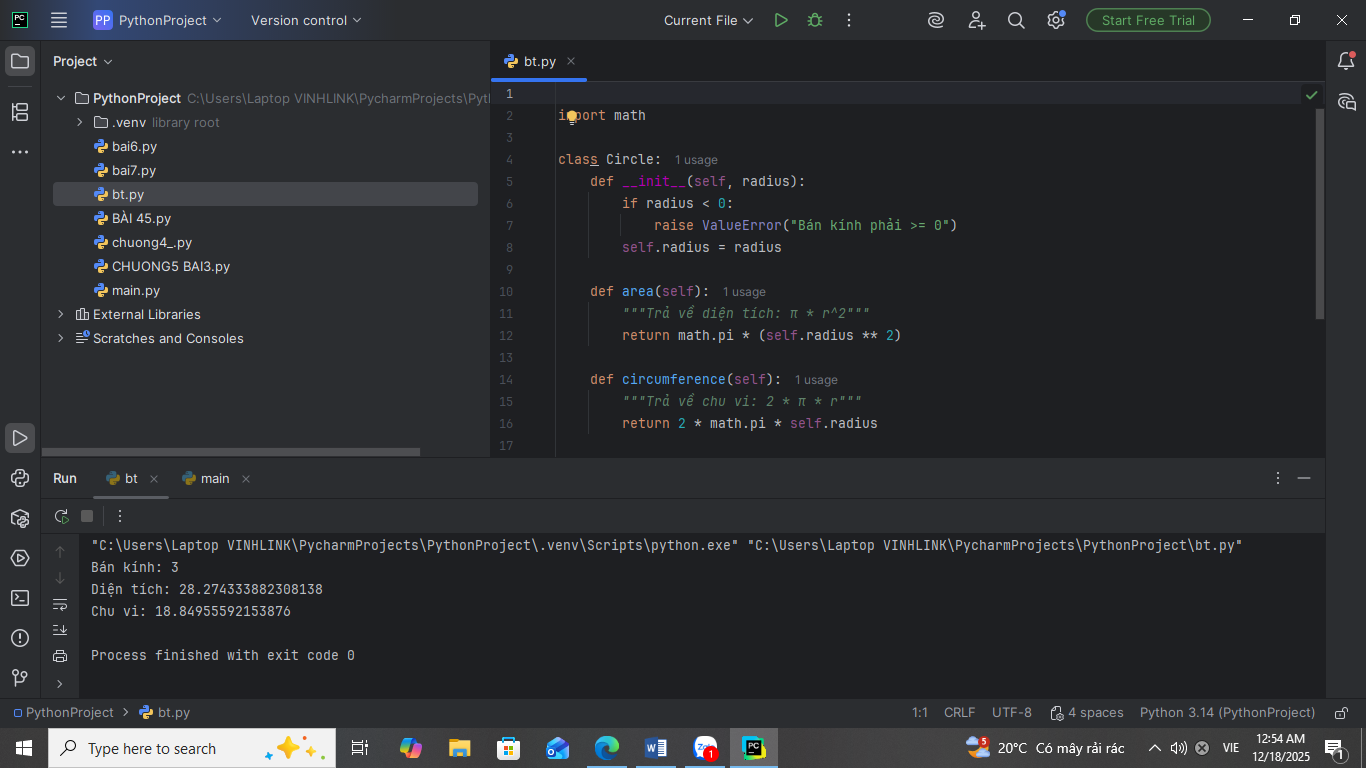
Công thức: circumference = 2 \* π \* r.

Kết quả: Trả về chu vi dạng số thực. (Với r=3 → ≈ 18.8495559215)

5. Tạo đối tượng và gọi phương thức

Ví dụ: c = Circle(3) → gọi c.area(), c.circumference().

Kết quả: In ra bán kính, diện tích và chu vi như nêu ở trên.



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

Câu 7.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. Nhập đường dẫn tệp

Người dùng nhập path (chuỗi) từ bàn phím.

Kết quả: có đường dẫn tệp để mở và đếm dòng.

2. Mở tệp an toàn

Dùng with open(path, 'r', encoding='utf-8') để mở tệp trong chế độ đọc; with đảm bảo tệp được đóng tự động.

Kết quả: tệp được mở (hoặc ném ngoại lệ nếu không tìm thấy/không có quyền).

3. Đếm dòng

Duyệt từng dòng trong file (for \_ in f:) và tăng bộ đếm count += 1.

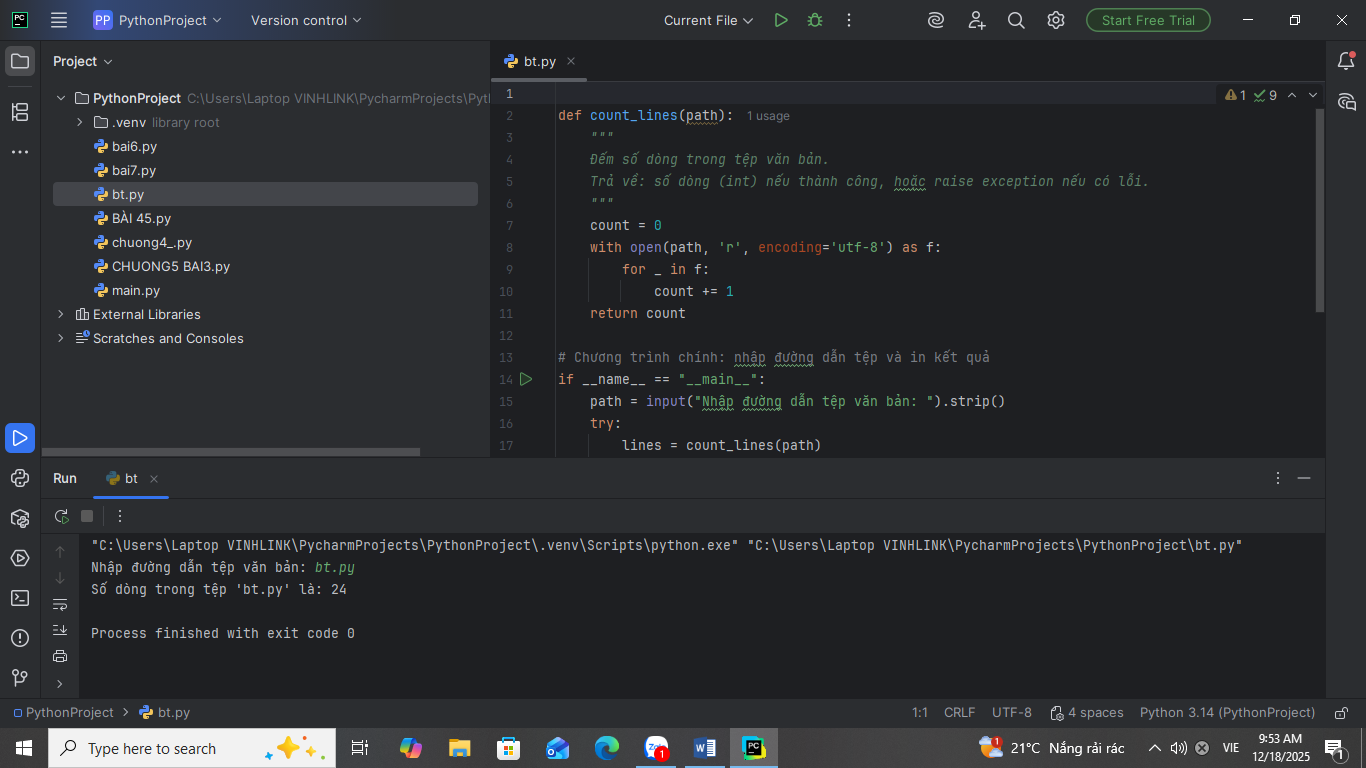
Kết quả: sau vòng lặp, count chứa tổng số dòng (bao gồm cả dòng rỗng).

4. Trả về & hiển thị kết quả

Hàm count\_lines(path) trả count (int).

Chương trình chính in: "Số dòng trong tệp 'path' là: <count>".

Kết quả: người dùng thấy số dòng tệp; nếu có lỗi (FileNotFound, PermissionError, ...), chương trình in thông báo lỗi phù hợp.



Câu 9.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. Nhập đường dẫn tệp nguồn và tệp đích

Người dùng nhập source và dest.

Kết quả: có hai đường dẫn để thực hiện thao tác sao chép.

2. Mở và đọc nội dung tệp nguồn

Dùng with open(source, 'r', encoding='utf-8') để mở an toàn.

Đọc toàn bộ nội dung bằng read().

Kết quả: lấy được dữ liệu của tệp nguồn (hoặc báo lỗi nếu không mở được).

3. Mở tệp đích và ghi nội dung

Dùng with open(dest, 'w', encoding='utf-8') để ghi đè vào tệp đích.

Ghi dữ liệu đọc được sang tệp mới.

Kết quả: nội dung tệp đích trở thành bản sao đúng của tệp nguồn.

4. Xử lý lỗi

Kiểm tra các ngoại lệ:

-Không tìm thấy tệp nguồn

-Không có quyền truy cập

-Lỗi khác trong quá trình đọc/ghi

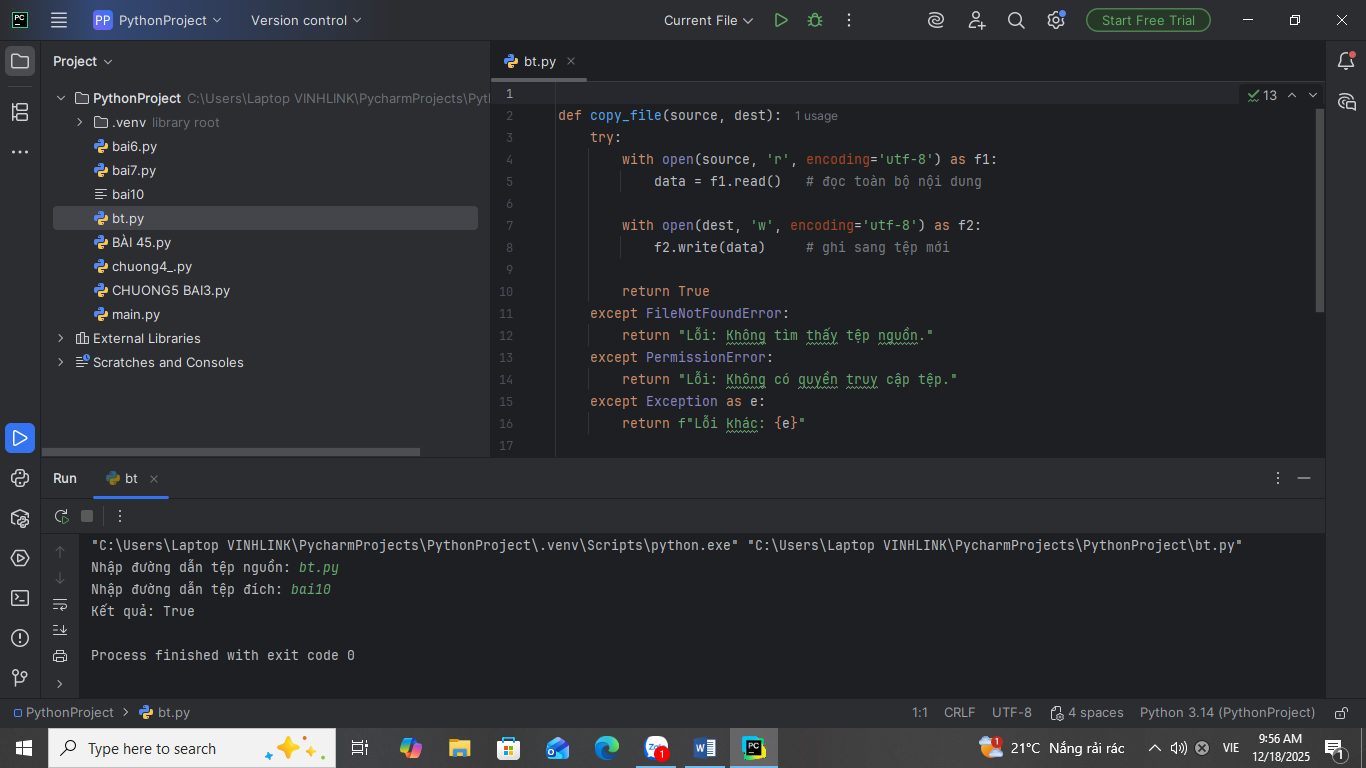
Kết quả: chương trình trả về thông báo lỗi rõ ràng nếu gặp sự cố.

5. Thông báo kết quả

Nếu không có lỗi → trả về True (thành công).

Nếu có lỗi → trả về chuỗi mô tả lỗi.

Kết quả: người dùng biết tệp đã sao chép thành công hay thất bại.



Câu 10.

I.Mục đích

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

II.Các bước thực hiện và kết quả

1. Nhập văn bản

Người dùng nhập chuỗi văn bản từ bàn phím.

Kết quả: có dữ liệu đầu vào để phân tích.

2. Tách từ và làm sạch

Dùng text.split() để tách thành danh sách từ.

Loại ký tự đặc biệt ở đầu/cuối mỗi từ bằng strip(...).

Kết quả: thu được danh sách từ “sạch” để xử lý chính xác.

3. Xác định độ dài lớn nhất

Tìm độ dài lớn nhất bằng max(len(w) for w in words).

Kết quả: có max\_len – độ dài của từ dài nhất trong văn bản.

4. Lọc các từ dài nhất

Lấy tất cả từ có độ dài bằng max\_len.

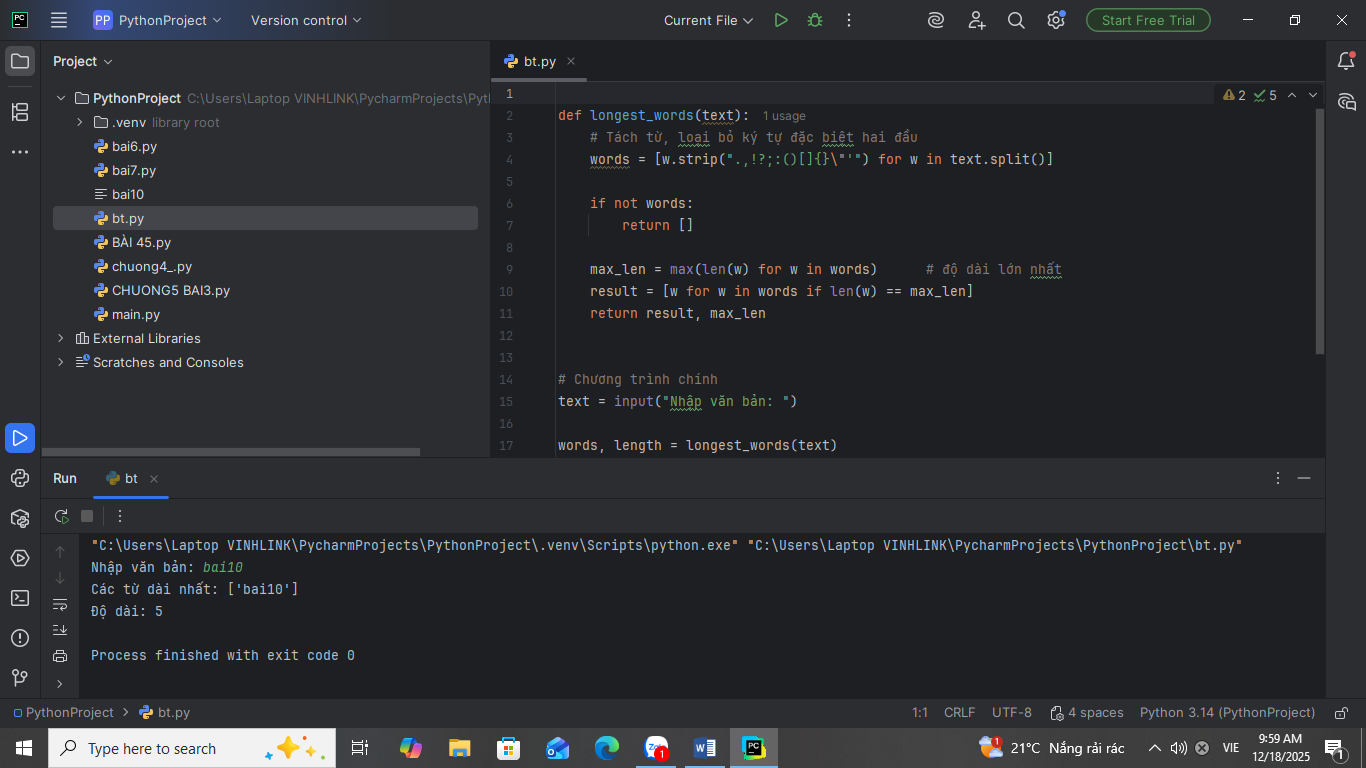
Kết quả: danh sách các từ dài nhất (có thể nhiều hơn 1).

5. Trả và in kết quả

Hàm trả danh sách từ dài nhất và độ dài của chúng.

In kết quả ra màn hình.

Kết quả: người dùng thấy được các từ dài nhất và độ dài tương ứng.



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

Câu 8a.

I.Mục đích

. 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)

II.Các bước thực hiện và kết quả

1.Tạo cửa sổ tkinter

Dùng tk.Tk() và đặt tiêu đề.

2. Chuẩn bị dữ liệu thông tin

Lưu vào dictionary: họ tên, ngày sinh, MSSV, ngành học.

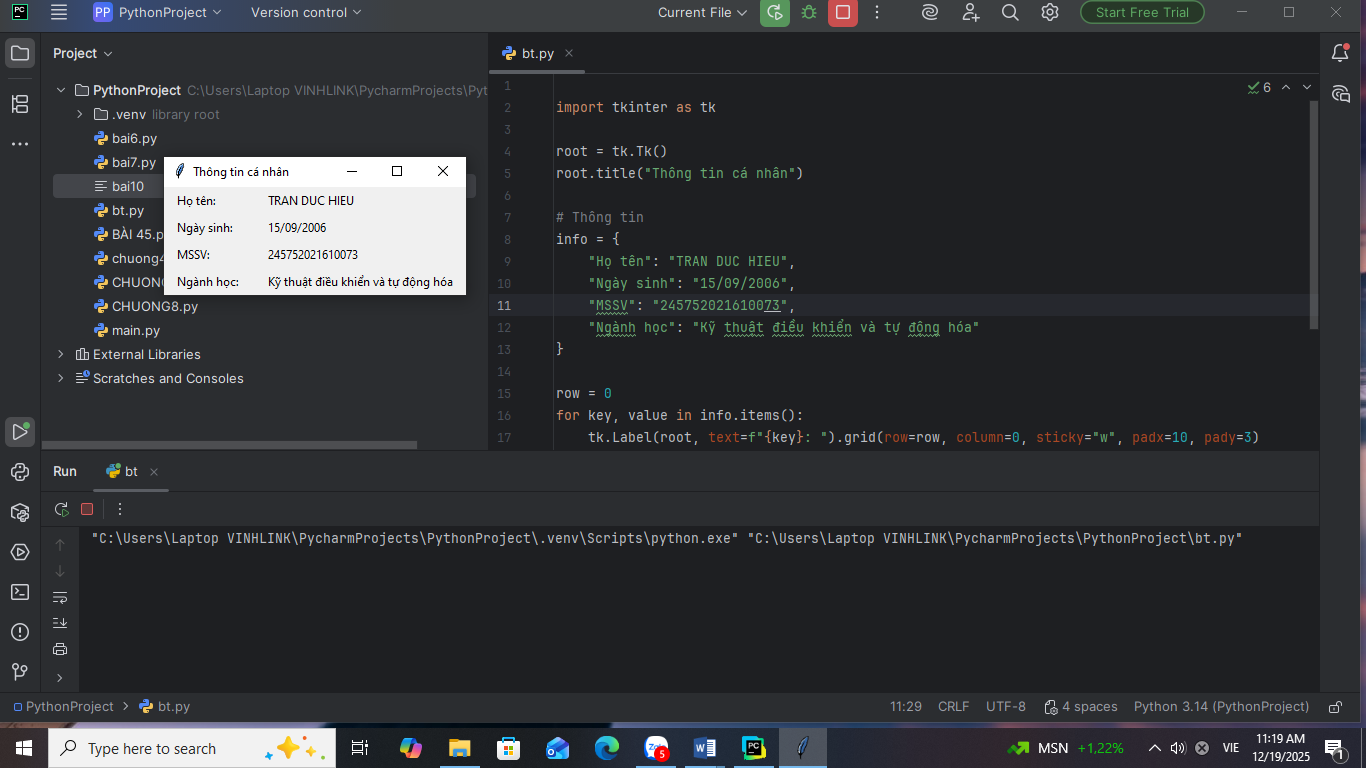
3. Hiển thị thông tin bằng Label

Dùng vòng lặp tạo từng dòng thông tin lên form.

4. Chạy vòng lặp sự kiện

root.mainloop() để hiển thị cửa sổ

Kết quả : Form hiển thị đầy đủ các thông tin cá nhân dưới dạng nhãn.



Câu 8b.

I.Mục đích

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)

II.Các bước thực hiện và kết quả

1.Tạo cửa sổ Tkinter

tk.Tk()

2. Tạo biến lựa chọn radio

value = tk.IntVar() để nhận giá trị 1, 2, 3.

3. Tạo 3 Radio Buttons

Mỗi nút gắn value=1, 2, 3.

4. Tạo nút "Click Me“

Khi bấm, hàm show\_selected() hiển thị lựa chọn.

5. Hiển thị cửa sổ

mainloop().

Kết quả:Người dùng chọn một trong 3 nút (First, Second, Third).Khi bấm “Click Me”, thông báo hiện ra giá trị tương ứng 1 – 2 – 3.

