A black background with white text

AI-generated content may be incorrect.

**Name: Quratulain sarwar**

**Intern ID:** **TN/IN02/PY/025**

**Task no: 2**

**Question: .**

**Store 5 student names & print each.**

**2. Reverse list without reverse().**

**CODE:**

students = ["Hadi", "Saim", "Hassan", "Ahmed", "Ali"]

# Print each student name

print("Student Names:")

for name in students:

    print(name)

# Step 2: Reverse list without using reverse()

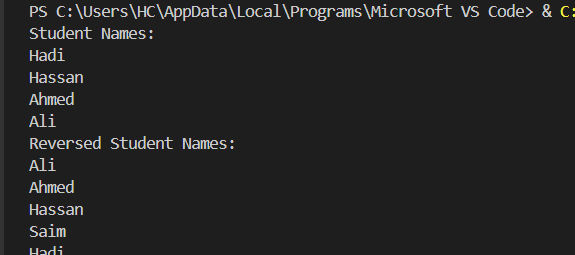
reversed\_students = students[::-1]

print("\nReversed Student Names:")

for name in reversed\_students:

    print(name)

**OUTPUT:**

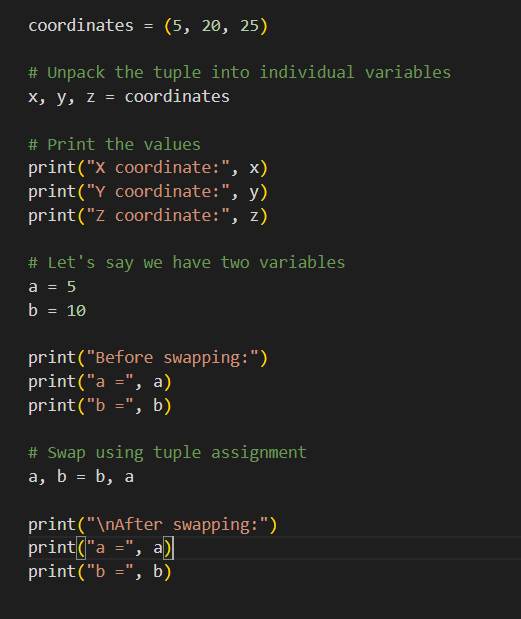


**QUESTION 2:**

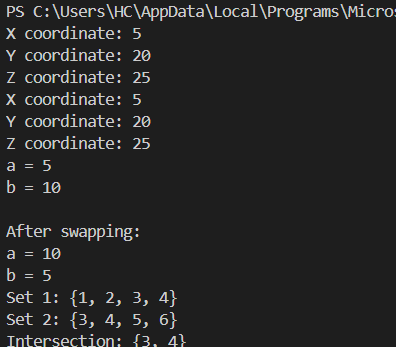
**. Store 3 coordinates & unpack.**

**2. Swap vars using tuple assignment**

**CODE:**



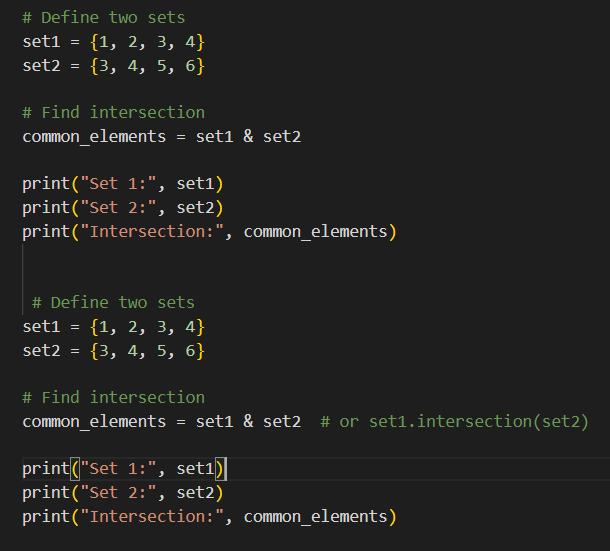
**OUTPUT:**



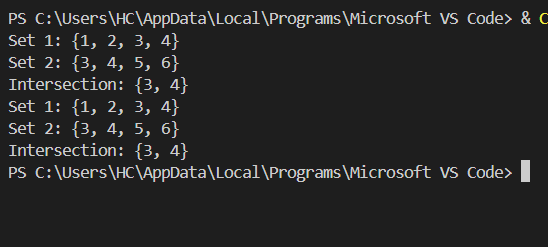
**QUESTION 3:**

**. Remove duplicates from list.**

**2. Find intersection of two sets.**



**OUTPUT:**

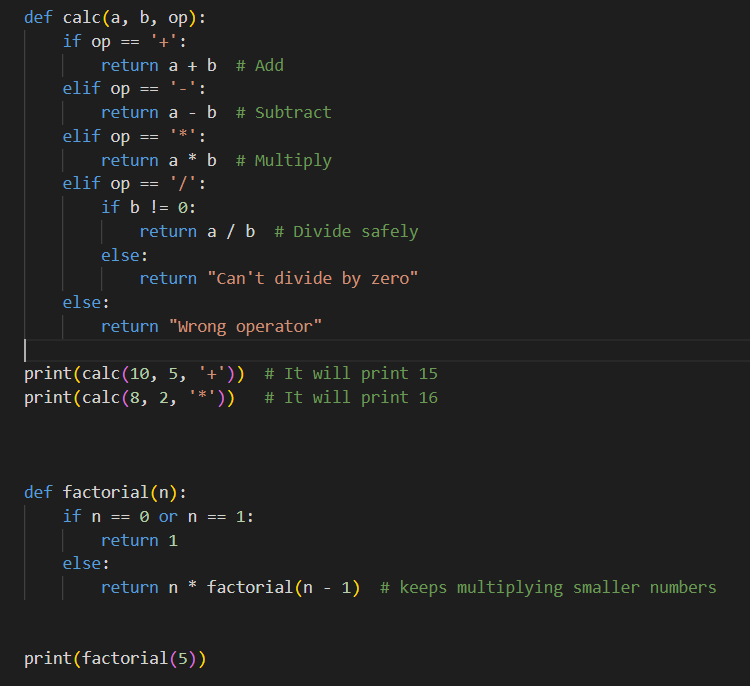


**QUESTION 4:**

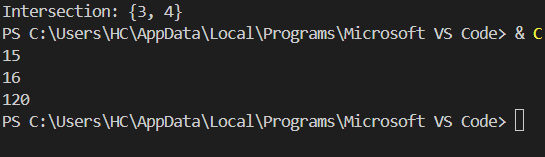
**. Student record CRUD in dict.**

**2. Count word frequency in sentence.**

**CODE:**



**OUTPUT:**

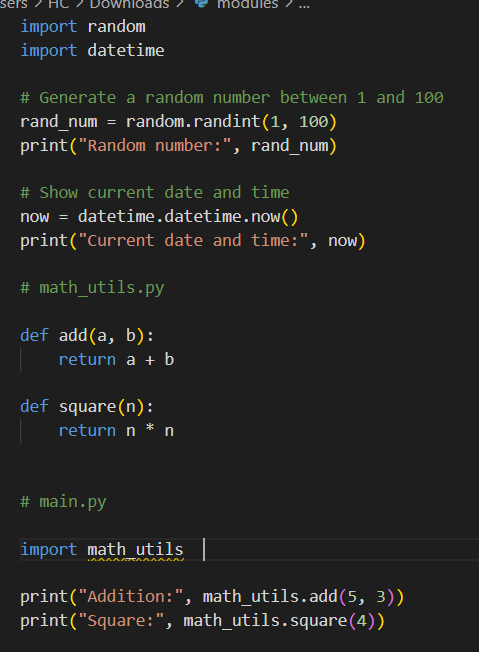


**QUESTION 5:**

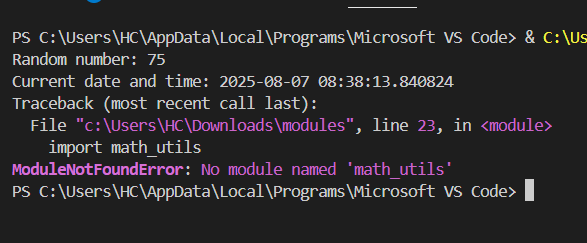
**. . Use random & datetime in script.**

**2. Create math\_utils module & import**

**CODE:**



**OUTPUT:**

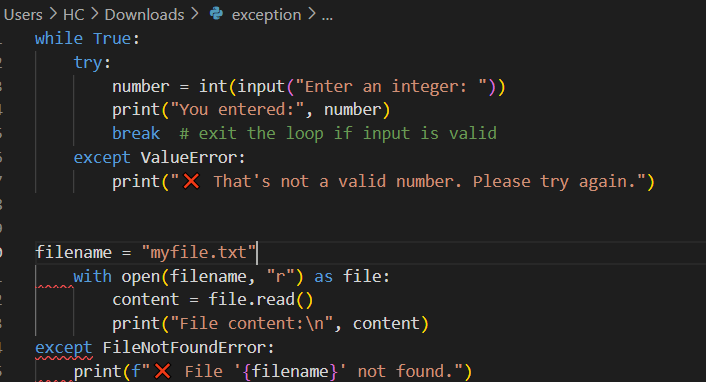
****

**QUESTION 6:**

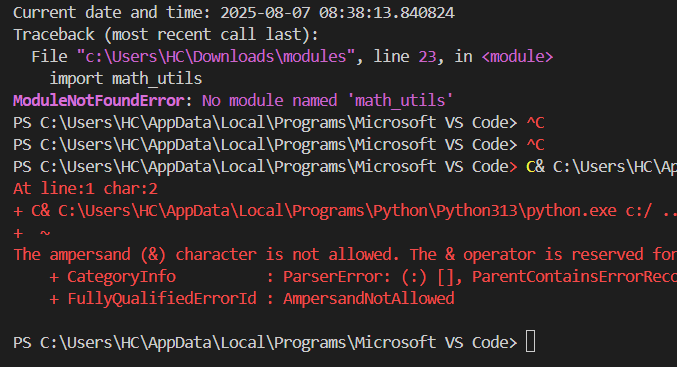
**. Safe int input loop.**

**2. File open with error message.**

**CODE:**

****

**OUTPUT:**

****

**Tasks Covered**

1. Take integer input safely using try-except loop

2. Open a file and handle file not found error

3. Create a calculator function `calc(a, b, op)`

4. Recursive factorial function

5. Student record CRUD using dictionary

6. Word frequency counter in a sentence

7. Remove duplicates from list using set

8. Find intersection of two sets

9. Unpack coordinates and swap variables

10. Use `random` and `datetime` modules

11. Create and import custom module (`math\_utils`)

12. Upload all code to GitHub step by step

**What I Learned**

- How to take safe user input using a `while` loop and `try-except`

- How to read a file and avoid crashing the program when the file doesn't exist

- Writing clean functions and reusing code

- Using recursion for mathematical problems

- Working with sets, lists, and dictionaries in Python

- Creating and importing custom Python modules

- Uploading code to GitHub using the web interface

**Difficulties I Faced**

**Handling exceptions:**

Understanding where to place `try` and `except` blocks took some time, especially during file handling. For example, I mistakenly placed the `except` block under the `with` statement, which caused a syntax error.

**Safe input loop**:

It was tricky to catch user input errors like entering letters instead of numbers, but using `ValueError` in `except` helped solve that.

**Recursion logic:**

In the factorial function, it was confusing at first to understand how the function calls itself until the base case.

**GitHub uploads**:

I was unsure whether I could upload an entire folder directly or only files. I learned that using GitHub's web interface, you can only upload files, not folders.

**Module creation**:

Creating and importing custom modules required understanding Python file structures and import statements.