AI ASSISTED CODING

Name: S.VYSHNAVI

Ht.No: 2303A52239

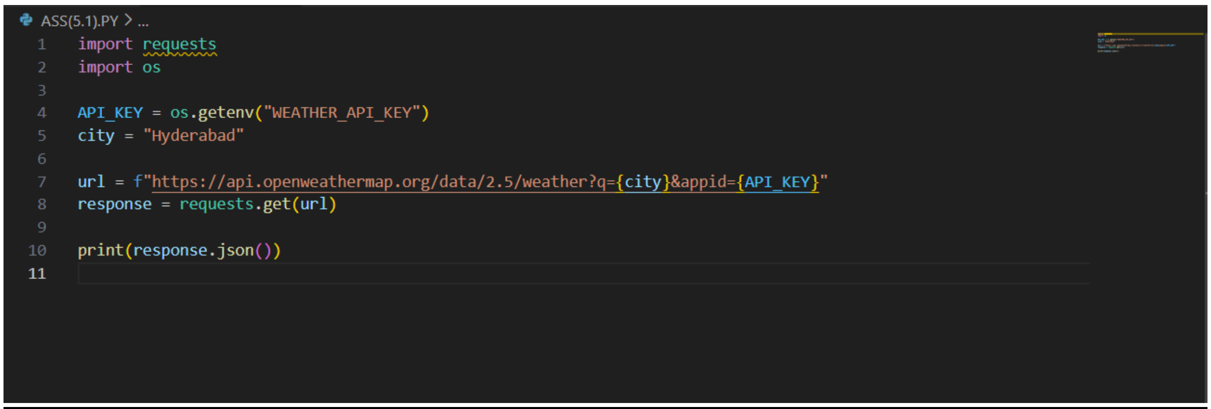
Batch:35

Assignment-5.1

Task-1: Use an AI tool to generate a Python program that connects to a weather API.

PROMPT: Generate a Python program that fetches current weather data from a public weather API. The API key should NOT be hardcoded in the program. Use environment variables to access the API key securely and explain why this method is safer.

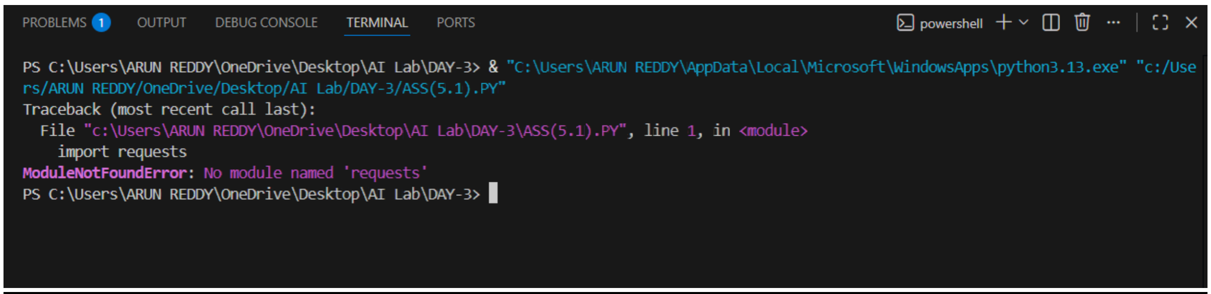
CODE:



ASS(5.1).PY > ...

```python
import requests
import os

API_KEY = os.getenv("WEATHER_API_KEY")
city = "Hyderabad"

url = f"https://api.openweathermap.org/data/2.5/weather?q={city}&appid={API_KEY}"
response = requests.get(url)

print(response.json())
```

OUTPUT:



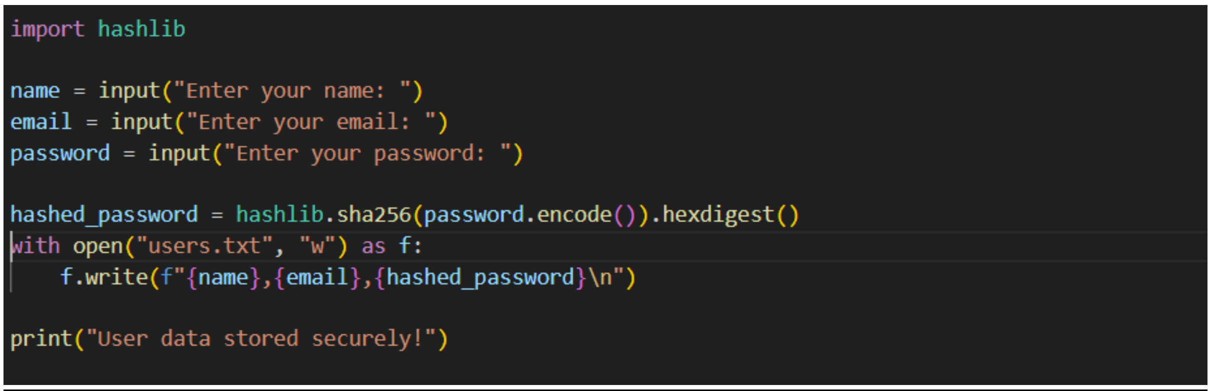
```
PS C:\Users\ARUN REDDY\OneDrive\Desktop\AI Lab\DAY-3> & "C:\Users\ARUN REDDY\AppData\Local\Microsoft\WindowsApps\python3.13.exe" "c:/Users/ARUN REDDY/OneDrive/Desktop/AI Lab/DAY-3/ASS(5.1).PY"
Traceback (most recent call last):
  File "c:\Users\ARUN REDDY\OneDrive\Desktop\AI Lab\DAY-3\ASS(5.1).PY", line 1, in <module>
    import requests
ModuleNotFoundError: No module named 'requests'
PS C:\Users\ARUN REDDY\OneDrive\Desktop\AI Lab\DAY-3>
```

TASK-2: Use an AI tool to generate a Python script that stores user data (name, email, password) in a file.

PROMPT:

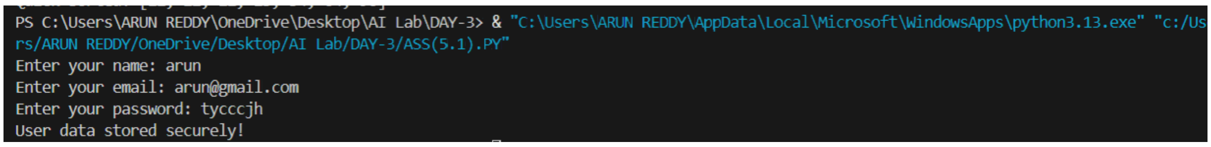
Generate a Python program that stores user data such as name, email, and password into a file. After generating the code, analyze whether storing passwords in plain text is safe. Then provide a secure version using password hashing or encryption with proper explanation.

CODE:



```python
import hashlib

name = input("Enter your name: ")
email = input("Enter your email: ")
password = input("Enter your password: ")

hashed_password = hashlib.sha256(password.encode()).hexdigest()
with open("users.txt", "w") as f:
    f.write(f"{name},{email},{hashed_password}\n")

print("User data stored securely!")
```

OUTPUT:



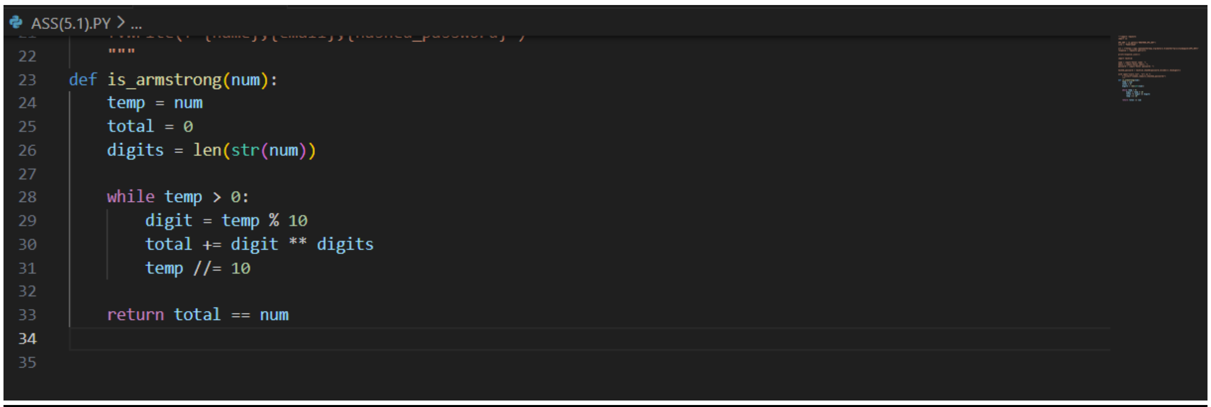
```
PS C:\Users\ARUN REDDY\OneDrive\Desktop\AI Lab\DAY-3> & "C:\Users\ARUN REDDY\AppData\Local\Microsoft\WindowsApps\python3.13.exe" "c:/Us
rs/ARUN REDDY/OneDrive/Desktop/AI Lab/DAY-3/ASS(5.1).PY"
Enter your name: arun
Enter your email: arun@gmail.com
Enter your password: tycccjh
User data stored securely!
```

TASK-3: Use AI to generate an Armstrong number checking function with comments and explanations.

PROMPT:

Generate a Python function to check whether a number is an Armstrong number. Add clear comments for every step in the code and explain each line in simple language so that a beginner can understand the algorithm.

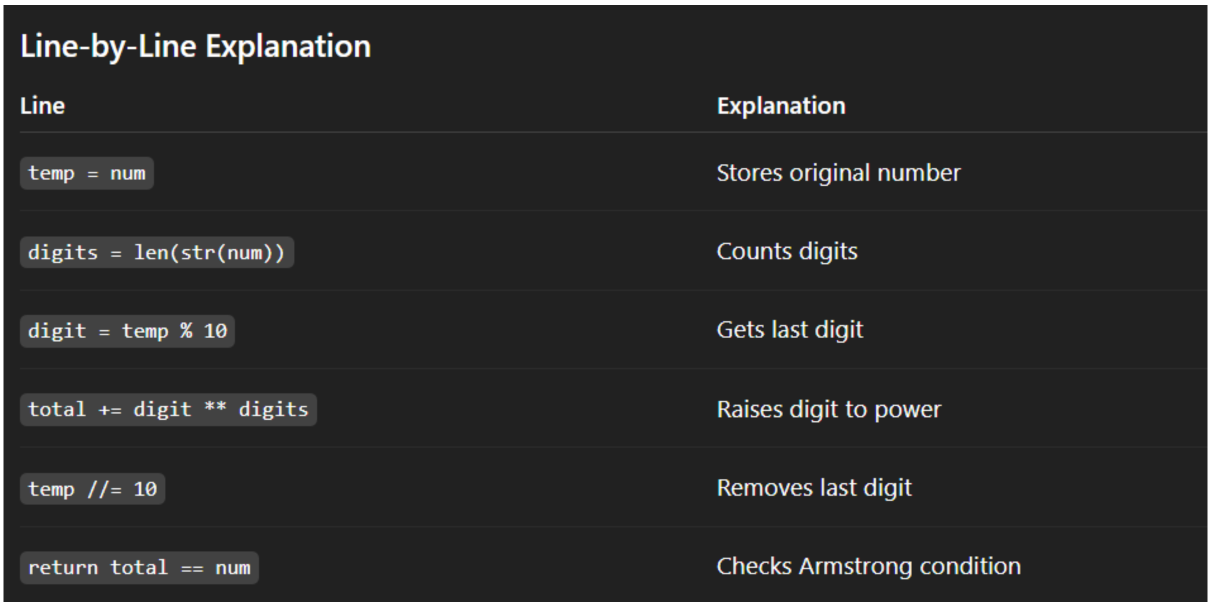
CODE:



ASS(5.1).PY > ...

```python
    """
def is_armstrong(num):
    temp = num
    total = 0
    digits = len(str(num))

    while temp > 0:
        digit = temp % 10
        total += digit ** digits
        temp //= 10

    return total == num
```

LINE-BY-LINE EXPLANATION:



Line-by-Line Explanation

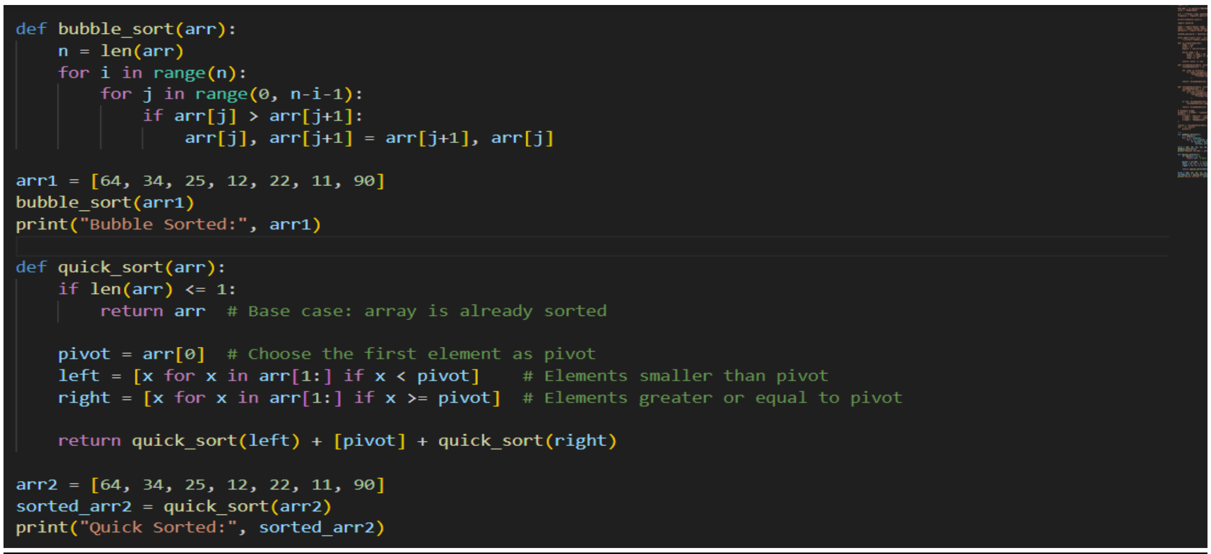
Line	Explanation
`temp = num`	Stores original number
`digits = len(str(num))`	Counts digits
`digit = temp % 10`	Gets last digit
`total += digit ** digits`	Raises digit to power
`temp //= 10`	Removes last digit
`return total == num`	Checks Armstrong condition

TASK-4: Use AI to implement two sorting algorithms (e.g., QuickSort and BubbleSort).

PROMPT:

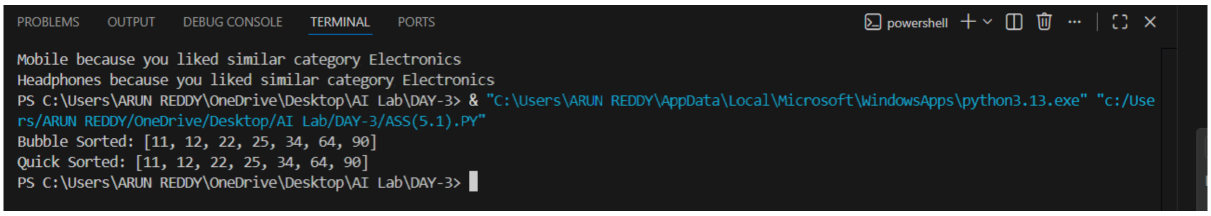
Generate Python code for Bubble Sort and Quick Sort algorithms. Add step-by-step comments explaining how each algorithm works. Then compare both algorithms in terms of logic, time complexity, and efficiency.

CODE:



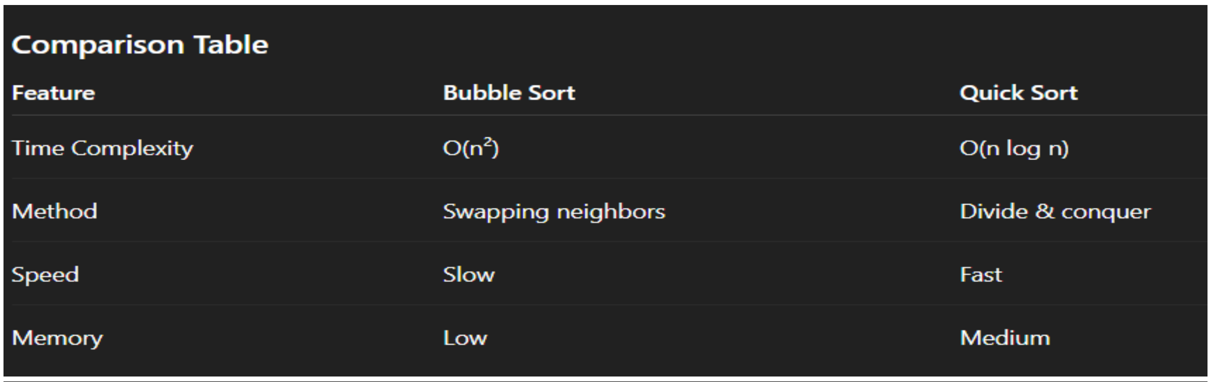
```python
def bubble_sort(arr):
    n = len(arr)
    for i in range(n):
        for j in range(0, n-i-1):
            if arr[j] > arr[j+1]:
                arr[j], arr[j+1] = arr[j+1], arr[j]

arr1 = [64, 34, 25, 12, 22, 11, 90]
bubble_sort(arr1)
print("Bubble Sorted:", arr1)

def quick_sort(arr):
    if len(arr) <= 1:
        return arr  # Base case: array is already sorted

    pivot = arr[0]  # Choose the first element as pivot
    left = [x for x in arr[1:] if x < pivot]    # Elements smaller than pivot
    right = [x for x in arr[1:] if x >= pivot]  # Elements greater or equal to pivot

    return quick_sort(left) + [pivot] + quick_sort(right)

arr2 = [64, 34, 25, 12, 22, 11, 90]
sorted_arr2 = quick_sort(arr2)
print("Quick Sorted:", sorted_arr2)
```

OUTPUT:



```
Mobile because you liked similar category Electronics
Headphones because you liked similar category Electronics
PS C:\Users\ARUN REDDY\OneDrive\Desktop\AI Lab\DAY-3> & "C:\Users\ARUN REDDY\AppData\Local\Microsoft\WindowsApps\python3.13.exe" "c:/Use
rs/ARUN REDDY/OneDrive/Desktop/AI Lab/DAY-3/ASS(5.1).PY"
Bubble Sorted: [11, 12, 22, 25, 34, 64, 90]
Quick Sorted: [11, 12, 22, 25, 34, 64, 90]
PS C:\Users\ARUN REDDY\OneDrive\Desktop\AI Lab\DAY-3> 
```

COMPARISON TABLE:



Comparison Table

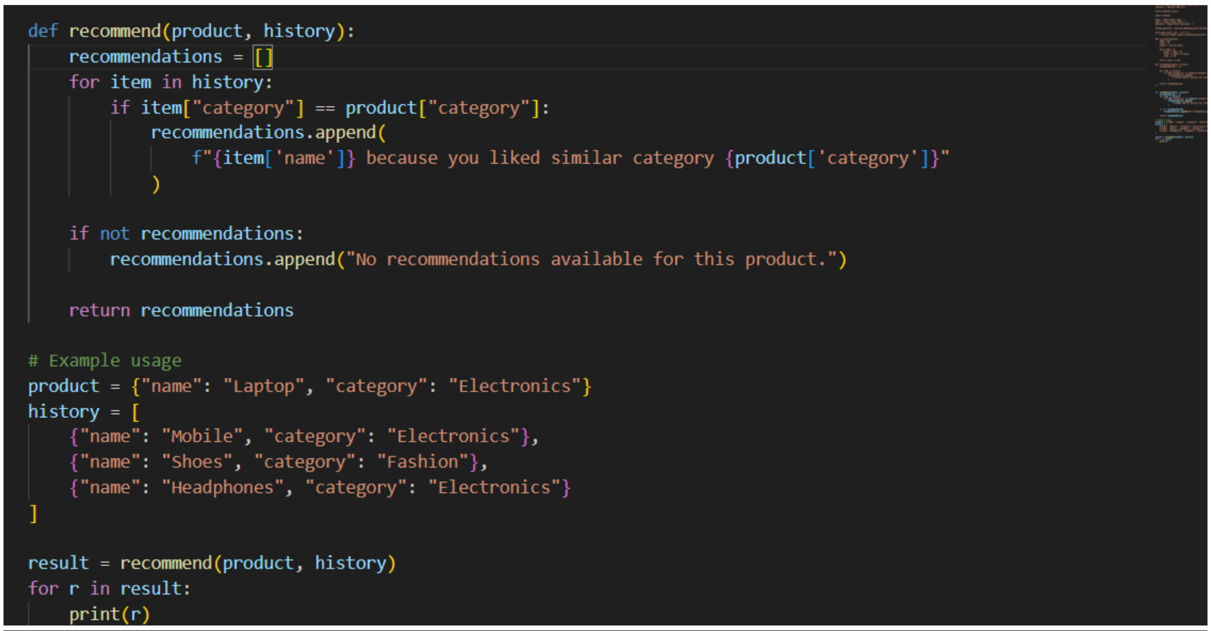
Feature	Bubble Sort	Quick Sort
Time Complexity	$O(n^2)$	O(n log n)
Method	Swapping neighbors	Divide & conquer
Speed	Slow	Fast
Memory	Low	Medium

TASK-5: Use AI to create a product recommendation system.

PROMPT:

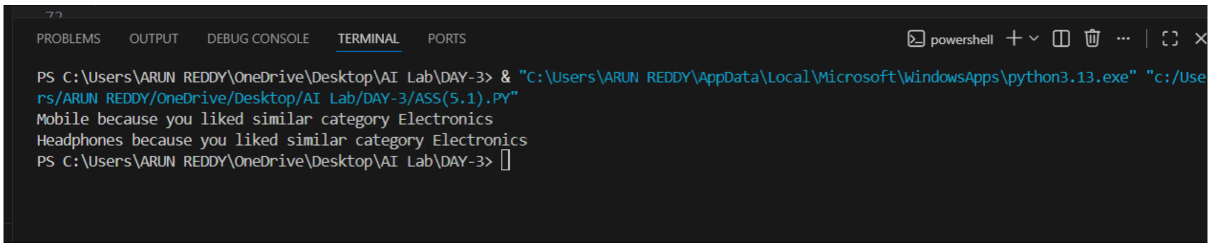
Generate a simple product recommendation system in Python. The system should not only recommend products but also clearly explain the reason behind each recommendation so the user can understand why the product was suggested.

CODE:



```python
def recommend(product, history):
    recommendations = []
    for item in history:
        if item["category"] == product["category"]:
            recommendations.append(
                f"{item['name']} because you liked similar category {product['category']}"
            )

    if not recommendations:
        recommendations.append("No recommendations available for this product.")

    return recommendations

# Example usage
product = {"name": "Laptop", "category": "Electronics"}
history = [
    {"name": "Mobile", "category": "Electronics"},
    {"name": "Shoes", "category": "Fashion"},
    {"name": "Headphones", "category": "Electronics"}
]

result = recommend(product, history)
for r in result:
    print(r)
```

OUTPUT:



```
PS C:\Users\ARUN REDDY\OneDrive\Desktop\AI Lab\DAY-3> & "C:\Users\ARUN REDDY\AppData\Local\Microsoft\WindowsApps\python3.13.exe" "c:/Use
rs/ARUN REDDY/OneDrive/Desktop/AI Lab/DAY-3/ASS(5.1).PY"
Mobile because you liked similar category Electronics
Headphones because you liked similar category Electronics
PS C:\Users\ARUN REDDY\OneDrive\Desktop\AI Lab\DAY-3>
```