Green University of Bangladesh

Department of Computer Science and Engineering(CSE)

Faculty of Sciences and Engineering

Semester: (Spring, Year:2025), B.Sc. in CSE (Day)

Course Code: CSE-316 Section: D14

Student Details

Name	ID
Ashraful Islam Miraj	221902231

Lab Date : 05-02-2025

Submission Date : 11-02-2025

Course Teacher's Name :Md. Sabbir Hosen Mamun

[For Teachers use only: Don't Write Anything inside this box]

Lab Report Status

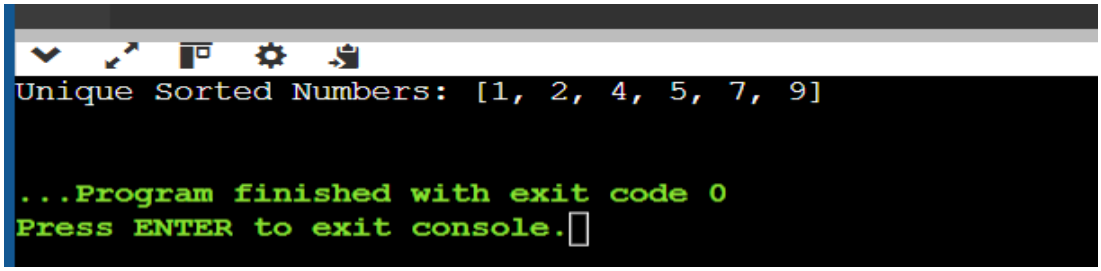
Marks: …… Signature:.....

Comments:..... Date:.....

1. List: Remove duplicates and sort in ascending order

CODE:

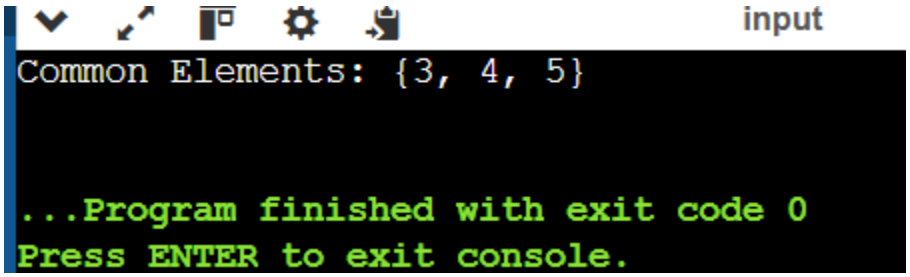
```
numbers = [4, 2, 7, 2, 5, 7, 9, 1]
unique_sorted_numbers = sorted(set(numbers))
print("Unique Sorted Numbers:", unique_sorted_numbers)
```



2. Set: Find common elements between two lists using sets

CODE:

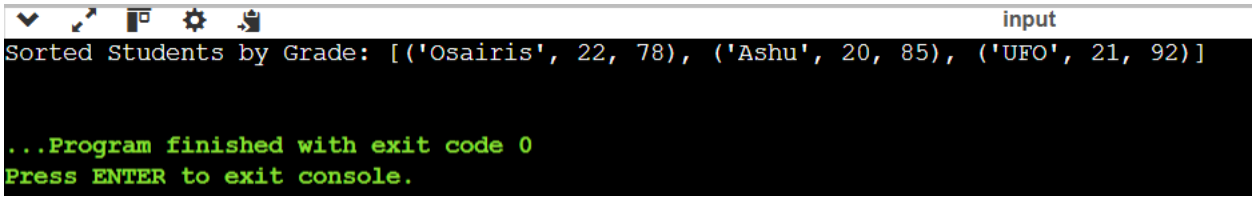
```
list1 = {1, 2, 3, 4, 5}
list2 = {3, 4, 5, 6, 7}
common_elements = list1.intersection(list2)
print("Common Elements:", common_elements)
```



3. Tuple: Create a tuple of student records and sort by grade

CODE:

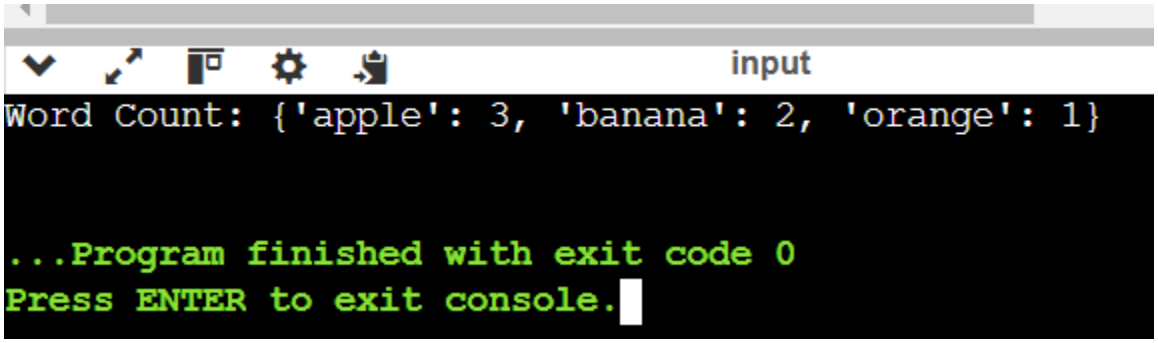
```
students = [("Ashu", 20, 85), ("Osairis", 22, 78), ("UFO", 21, 92)]
sorted_students = sorted(students, key=lambda x: x[2])
print("Sorted Students by Grade:", sorted_students)
```



4. Dictionary: Count word occurrences in a given text

CODE:

```
text = "apple banana apple orange banana apple"
words = text.split()
word_count = {word: words.count(word) for word in set(words)}
print("Word Count:", word_count)
```



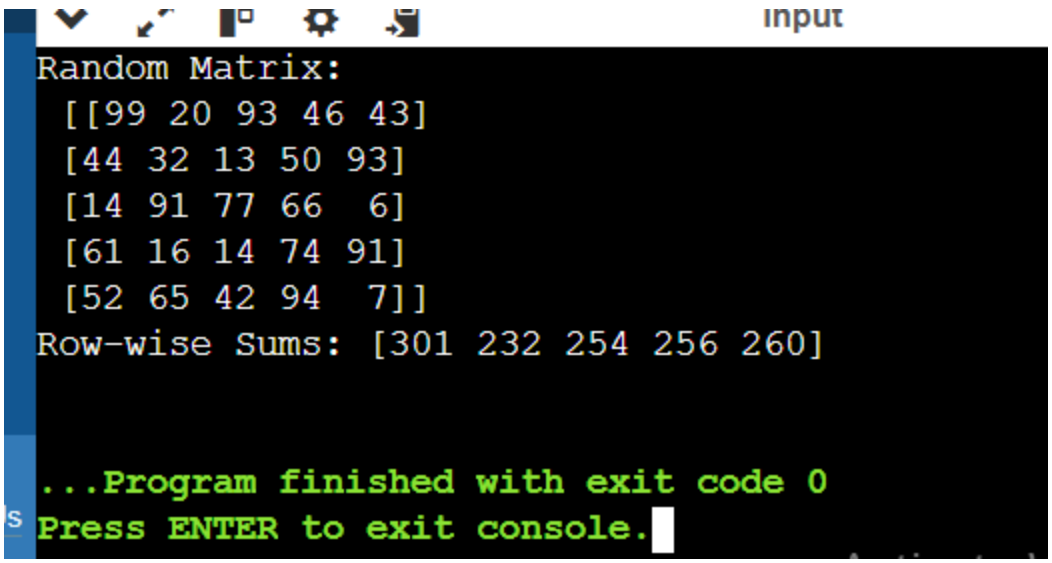
5. NumPy#1: Generate a 5x5 matrix of random integers and compute row-wise sums.

CODE:

```
import numpy as np

matrix = np.random.randint(1, 100, (5, 5))
row_sums = matrix.sum(axis=1)

print("Random Matrix:\n", matrix)
print("Row-wise Sums:", row_sums)
```

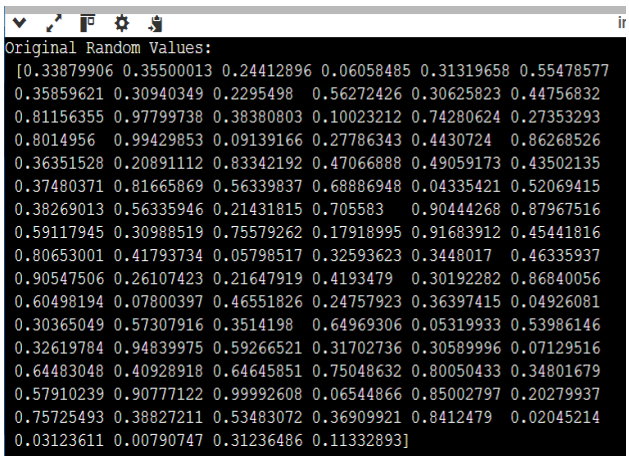


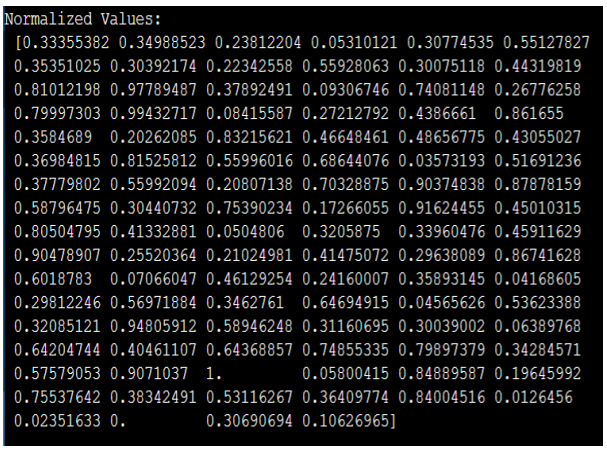
6. NumPy#2: Create an array of 100 random values and normalize them between 0 and 1

CODE:

```
import numpy as np
random_values = np.random.rand(100)
normalized_values = (random_values - np.min(random_values)) /
(np.max(random_values) - np.min(random_values))

print("Original Random Values:\n", random_values)
print("\nNormalized Values:\n", normalized_values)
```



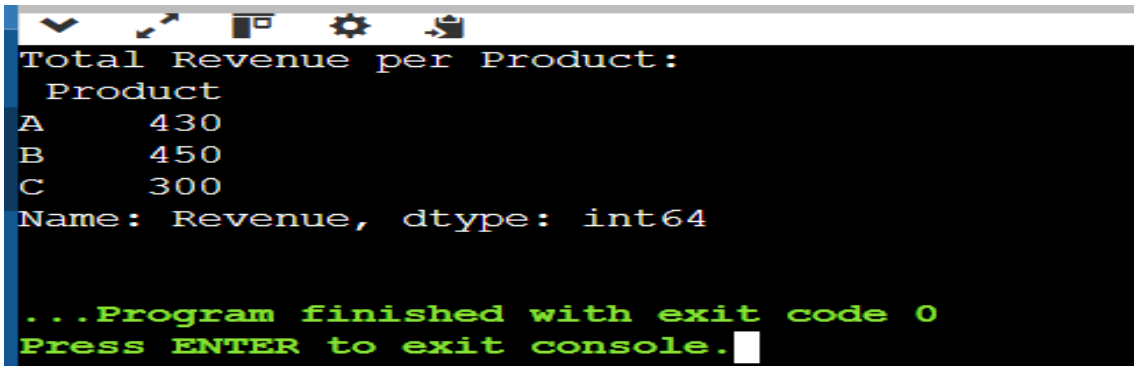


7. Pandas#1: Load a CSV file of sales data and compute total revenue per product

CODE:

```
import pandas as pd
data = {
    'Product': ['A', 'B', 'A', 'C', 'B', 'A'],
    'Revenue': [100, 200, 150, 300, 250, 180]
}
sales_df = pd.DataFrame(data)
total_revenue = sales_df.groupby('Product')['Revenue'].sum()

print("Total Revenue per Product:\n", total_revenue)
```



8. Pandas#2: Fill missing values in a dataset with column-wise means

CODE:

```
import pandas as pd
import numpy as np
data = {'A': [1, np.nan, 3], 'B': [4, 5, np.nan]}
df = pd.DataFrame(data)
df_filled = df.fillna(df.mean())
print("Original DataFrame with Missing Values:\n", df)
print("\nDataFrame after Filling Missing Values:\n", df_filled)
```

```
Original DataFrame with Missing Values:
     A    B
0  1.0  4.0
1  NaN  5.0
2  3.0  NaN

DataFrame after Filling Missing Values:
     A    B
0  1.0  4.0
1  2.0  5.0
2  3.0  4.5


...Program finished with exit code 0
```

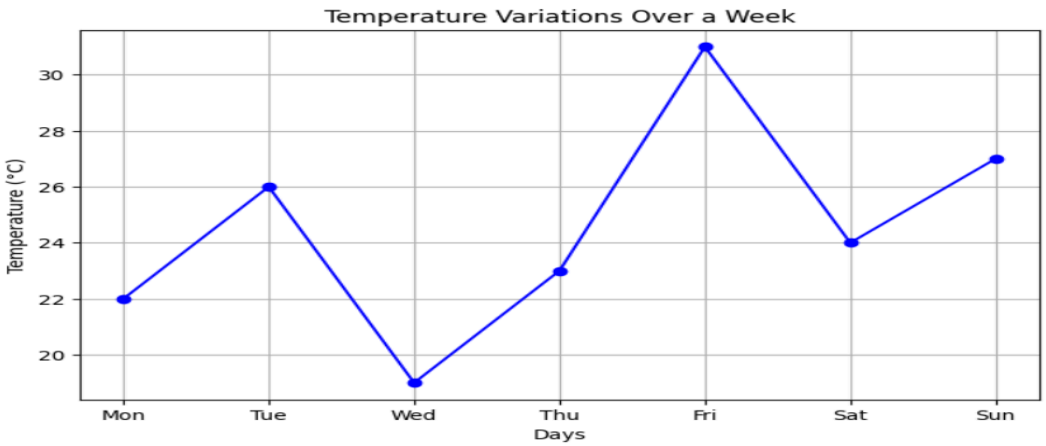
9. Matplotlib#1: Plot a line graph showing temperature variations over a week

code:

```
import matplotlib.pyplot as plt

days = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
temperatures = [22, 24, 19, 23, 25, 21, 20]

plt.figure(figsize=(8,5))
plt.plot(days, temperatures, marker='o', linestyle='-', color='b')
plt.xlabel("Days")
plt.ylabel("Temperature (°C)")
plt.title("Temperature Variations Over a Week")
plt.grid(True)
plt.show()
```



10. Matplotlib#2: Create a bar chart comparing sales revenue across different regions

CODE:

```
import matplotlib.pyplot as plt
regions = ["North", "South", "East", "West"]
sales_revenue = [25000, 30000, 22000, 27000]
plt.figure(figsize=(8,5))
plt.bar(regions, sales_revenue, color=['blue', 'red', 'green',
'purple'])
plt.xlabel("Regions")
plt.ylabel("Sales Revenue ($)")
plt.title("Sales Revenue Across Regions")
plt.grid(axis='y', linestyle='--')
plt.show()
```