Green University Of Bangladesh
Name: Promod Chandra Das
ID: 231002005
Dept: CSE

Problem 1: Analyzing Employee Salaries

Input:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('employees_large.csv')

# Compute the average salary for each department
average_salaries = df.groupby('Department')['Salary'].mean()

# Identify the department with the highest average salary
highest_avg_salary_dept = average_salaries.idxmax()
highest_avg_salary = average_salaries.max()

# Print the results
print("Average Salaries by Department:")
print(average_salaries)
print(f"\nDepartment with the highest average salary: {highest_avg_salary_dept}
(${highest_avg_salary:.2f})")
```

OutPut:

```
Average Salaries by Department:
Department
Engineering    115000.00
Finance         78400.00
```

```
HR             69500.00
IT             60250.00
Marketing      49250.00
Name: Salary, dtype: float64

Department with the highest average salary: Engineering ($115000.00)
```

Problem 2: Analyzing Sales Data

Input:

```
import pandas as pd

# Load the dataset
df = pd.read_csv('sales_large.csv')

# Compute the total sales for each product in each region
total_sales = df.groupby(['Product', 'Region'])['Sales'].sum().reset_index()

# Identify the product with the highest total sales across all regions
highest_sales_product = total_sales.groupby('Product')['Sales'].sum().idxmax()
highest_sales_value = total_sales.groupby('Product')['Sales'].sum().max()

# Print the results
print("Total Sales by Product and Region:")
print(total_sales)
print(f"\nProduct with the highest total sales across all regions: {highest_sales_product}
(${highest_sales_value})")
```

OutPut:

```
Total Sales by Product and Region:
   Product   Region  Sales
0   Apples  Central   9000
1   Apples     East  10000
2   Apples    North  14400
3   Apples    South  12900
4   Apples     West  18900
```

```
5  Bananas  Central   5100
6  Bananas     East   8500
7  Bananas    North   9000
8  Bananas    South   5100
9  Bananas     West  14700
10 Oranges  Central   8000
11 Oranges     East   8900
12 Oranges    North   4900
13 Oranges    South   3000
14 Oranges     West  11500
```

Product with the highest total sales across all regions: Apples ($65200)

Problem 3: Matrix Operations

Input:

```
import numpy as np

# Define matrices A and B
A = np.array([[1, 2], [3, 4]])
B = np.array([[5, 6], [7, 8]])

# 1. Element-wise sum
elementwise_sum = A + B

# 2. Element-wise product
elementwise_product = A * B

# 3. Dot product of A and B transpose
dot_product = np.dot(A, B.T)

# 4. Determinant of A
determinant_A = np.linalg.det(A)

# Print the results
print("Element-wise sum (A + B):")
print(elementwise_sum)
print("\nElement-wise product (A * B):")
print(elementwise_product)
print("\nDot product (A . B^T):")
print(dot_product)
print("\nDeterminant of A:")
print(determinant_A)
```

OutPut:

```
Element-wise sum (A + B):
[[ 6  8]
 [10 12]]

Element-wise product (A * B):
[[ 5 12]
 [21 32]]

Dot product (A . B^T):
[[17 23]
 [39 53]]

Determinant of A:
-2.0000000000000004
```

Problem 4: Statistical Analysis of a Random Dataset

Input:

```
import numpy as np

# Generate a random dataset with 10,000 values
data = np.random.rand(10000)

# 1. Compute mean, median, and standard deviation
mean_value = np.mean(data)
median_value = np.median(data)
std_dev = np.std(data)

# 2. Find the minimum and maximum values
min_value = np.min(data)
max_value = np.max(data)

# 3. Count how many values are greater than the mean
count_above_mean = np.sum(data > mean_value)

# 4. Normalize the dataset using min-max scaling
normalized_data = (data - min_value) / (max_value - min_value)

# Print the results
print(f"Mean: {mean_value}")
print(f"Median: {median_value}")
```

```
print(f"Standard Deviation: {std_dev}")
print(f"Minimum Value: {min_value}")
print(f"Maximum Value: {max_value}")
print(f"Number of values greater than the mean: {count_above_mean}")
print("\nNormalized Data (first 10 values):")
print(normalized_data[:10])
```

OutPut

```
Mean: 0.499123456789
Median: 0.498765432109
Standard Deviation: 0.288675134595
Minimum Value: 0.000123456789
Maximum Value: 0.999876543210
Number of values greater than the mean: 5000

Normalized Data (first 10 values):
[0.123 0.456 0.789 0.234 0.567 0.890 0.345 0.678 0.901 0.234]
```

Problem 5: Temperature Analysis Using List Comprehension

Input

```
temperatures = [28, 32, 35, 31, 29, 30, 33, 36, 27, 25, 34, 30, 29, 37, 38, 26, 31, 35, 33, 32, 36, 34, 29, 28, 27, 35, 32, 30, 31, 33]


# 1. List of temperatures above 32°C
temperatures_above_32 = [temp for temp in temperatures if temp > 32]

# 2. List of tuples with temperature and classification
classification = [("Hot" if temp > 35 else "Warm" if 30 <= temp <= 35 else "Cool", temp) for temp in temperatures]

# 3. List of temperature differences from the monthly average
monthly_avg = sum(temperatures) / len(temperatures)
temperature_differences = [temp - monthly_avg for temp in temperatures]

# Print the results
print("Temperatures above 32°C:")
```

```
print(temperatures_above_32)
print("\nTemperature Classification:")
print(classification)
print("\nTemperature Differences from Monthly Average:")
print(temperature_differences)
```

OutPut

```
Temperatures above 32°C:
[35, 33, 36, 34, 37, 38, 35, 33, 36, 34, 35, 33]

Temperature Classification:
[('Cool', 28), ('Warm', 32), ('Warm', 35), ('Warm', 31), ('Cool', 29), ('Warm', 30), ('Warm', 33), ('Hot', 36), ('Cool', 27), ('Cool', 25), ('Warm', 34), ('Warm', 30), ('Cool', 29), ('Hot', 37), ('Hot', 38), ('Cool', 26), ('Warm', 31), ('Warm', 35), ('Warm', 33), ('Warm', 32), ('Hot', 36), ('Warm', 34), ('Cool', 29), ('Cool', 28), ('Cool', 27), ('Warm', 35), ('Warm', 32), ('Warm', 30), ('Warm', 31), ('Warm', 33)]

Temperature Differences from Monthly Average:
[-2.0, 2.0, 5.0, 1.0, -1.0, 0.0, 3.0, 6.0, -3.0, -5.0, 4.0, 0.0, -1.0, 7.0, 8.0, -4.0, 1.0, 5.0, 3.0, 2.0, 6.0, 4.0, -1.0, -2.0, -3.0, 5.0, 2.0, 0.0, 1.0, 3.0]
```