Python DA Assignment 1: Data Analysis using NumPy and Pandas

Numpy Array Operations:

Problem Statement: To learn how to use NumPy for creating arrays, performing basic array operations, and conducting numerical computations through analyzing temperature data.

Scenario: You are given a list of daily average temperatures (in degrees Celsius) recorded over two weeks. Your task is to use NumPy to analyze this data.

Step wise Instructions:

1. **Create a 1D NumPy Array:**

   Create a 1D NumPy array named `temperatures_w1` for Week 1 with the following values:
   `[22.5, 25.3, 20.8, 23.4, 26.1, 24.8, 21.9]`

2. **Inspection and Properties:**
   - Inspect the shape, data type, and the number of elements in the array. Inspect the shape of the `temperatures_w1` array.

3. **Array Operations:**
   - Convert the `temperatures_w1` from Celsius to Fahrenheit using the formula: **`Fahrenheit = (Celsius * 9/5) + 32`**
   - Find the maximum, minimum, and mean temperatures for the week.

4. **Array Slicing and Indexing:**
   - Extract the temperatures for the first three days of the week.
   - Extract the temperatures for the weekend (last two days).
   - Extract the temperatures for the middle three days of the week.

5. **Create a 2D Array:**

   Create a 2D NumPy array named `temperatures` where each row represents temperatures for a different week:

   - ★ **Row 1:** `[22.5, 25.3, 20.8, 23.4, 26.1, 24.8, 21.9]` (Week 1)
   - ★ **Row 2:** `[19.2, 22.5, 21.3, 24.0, 23.5, 22.8, 20.1]` (Week 2)

6. **Inspect and Slice the 2D Array:**

   - Inspect and print the shape, data type, and total number of elements in the `temperatures` array.
   - Extract the temperatures for each week and the weekends (last two days) for both weeks.

Pandas Series:

1. Creating Pandas Series:

- Create a Pandas Series named `marks` with the following data:
  - ➢ **Marks:** `95, 92, 89, 85, 80`
  - ➢ **Custom Indices:** `'Rank1', 'Rank2', 'Rank3', 'Rank4', 'Rank5'`

2. Indexing and Slicing:

- Use the integer index position to access the mark of the 1st rank student.
- Use the `loc` accessor to retrieve the marks of the top 3 ranks by specifying their index labels.
- Use the `iloc` accessor to retrieve the mark of the 3rd rank student.
- Apply a boolean mask to filter the Series and retrieve the ranks where the marks are greater than `90`.

3. Manipulating Series:

- Modify the mark of the student with the 1st rank to `100`.
- Remove the entry corresponding to the last rank from the Series.
- Compute the CGPA by dividing each mark in the Series by 10.

Pandas DataFrame:

1. Creating Pandas DataFrame:

- Create a Pandas DataFrame named `transactions` with the following data:
  - **TransactionID:** 101, 102, 103, 104, 105, 106, 107, 108, 109, 110
  - **ProductCategory:** 'Electronics', 'Clothing', 'Electronics', 'Furniture', 'Clothing', 'Electronics', 'Furniture', 'Clothing', 'Furniture', 'Electronics'
  - **Region:** 'North', 'South', 'North', 'East', 'West', 'North', 'East', 'West', 'South', 'North'
  - **Amount:** 200, 150, 300, 450, 200, 250, 300, 180, 350, 400

2. Data Exploration:

- Display the `transactions` DataFrame and its basic information, including `head`, `tail`, shape, column names, and data types.
- Display only the *'ProductCategory'* and *'Amount'* columns.
- Retrieve the last 3 columns of the DataFrame using `iloc` or `loc`.
- Filter rows where the *'Region'* is 'North' and *'Amount'* is greater than 200.
- Find the value counts for the *'ProductCategory'* column.
- Find the unique values in the *'Region'* column.
- Group by *'Region'* and find the mean amount for each region.

3. Manipulating the DataFrame:

- Modify the *'Amount'* for the transaction with `TransactionID` `102` to 165.
- Add a new column *'Discount'* by calculating 10% of the *'Amount'*.
- Remove the row with `TransactionID` 109.
- Delete the *'Discount'* column from the DataFrame.