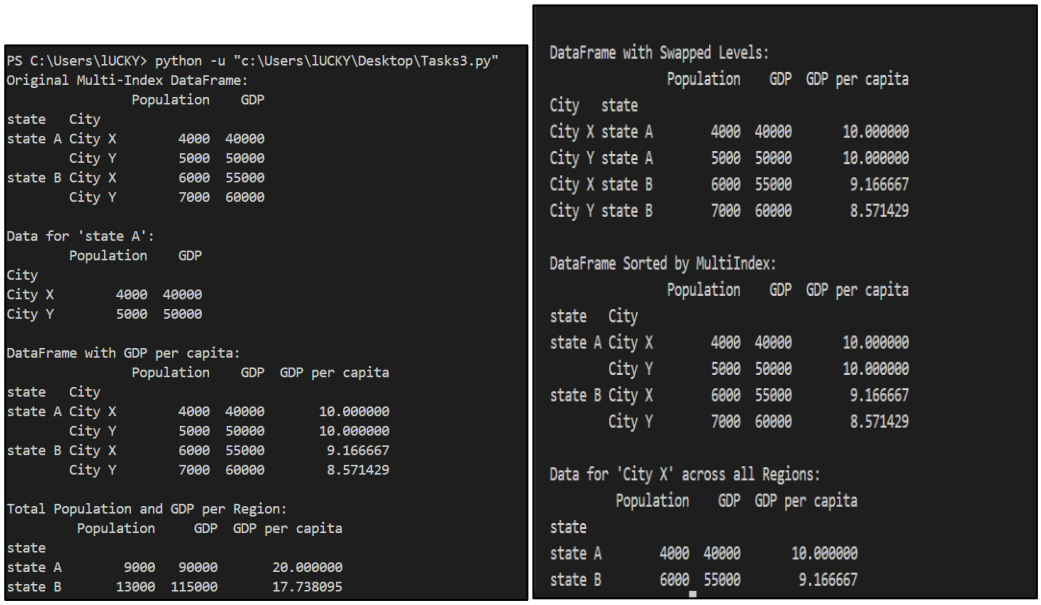
Task Report:

1. **Task Description**

- This task demonstrates how to create and manipulate a MultiIndex Data Frame in Python using **Pandas**. The key operations include:
- Creating a MultiIndex Data Frame from hierarchical data.
- Performing indexing and slicing operations.
- Adding a calculated column (GDP per capita).
- Grouping data by specific index levels to calculate aggregated values.
- Swapping and sorting index levels.
- Using cross-section (xs) to extract specific data based on index levels.

2. **Task Output Screenshot**



```
PS C:\Users\lUCKY> python -u "c:\Users\lUCKY\Desktop\Tasks3.py"
Original Multi-Index DataFrame:
               Population    GDP
state   City
state A City X        4000  40000
        City Y        5000  50000
state B City X        6000  55000
        City Y        7000  60000

Data for 'state A':
        Population    GDP
City
City X        4000  40000
City Y        5000  50000

DataFrame with GDP per capita:
               Population    GDP  GDP per capita
state   City
state A City X        4000  40000       10.000000
        City Y        5000  50000       10.000000
state B City X        6000  55000        9.166667
        City Y        7000  60000        8.571429

Total Population and GDP per Region:
        Population     GDP  GDP per capita
state
state A        9000   90000       20.000000
state B       13000  115000       17.738095
```

```
DataFrame with Swapped Levels:
               Population    GDP  GDP per capita
City   state
City X state A        4000  40000       10.000000
City Y state A        5000  50000       10.000000
City X state B        6000  55000        9.166667
City Y state B        7000  60000        8.571429

DataFrame Sorted by MultiIndex:
               Population    GDP  GDP per capita
state   City
state A City X        4000  40000       10.000000
        City Y        5000  50000       10.000000
state B City X        6000  55000        9.166667
        City Y        7000  60000        8.571429

Data for 'City X' across all Regions:
        Population    GDP  GDP per capita
state
state A        4000  40000       10.000000
state B        6000  55000        9.166667
```

3. **Widget/Algorithm Used In Task**
   Algorithm/Technique Used:

- The **twinx()** method in Matplotlib creates a second y-axis that shares the same x-axis as the original. This is useful for displaying two datasets with vastly different scales in one plot.
- The sine wave (y1 = sin(x)) represents periodic data, while the exponential function
- (y2 = exp(x/3)) represents rapid growth. Both were plotted using plot() function.