

## Questions related to Numpy and Pandas

In [50]:

```
1 #importing necessary libraries
2
3 import numpy as np
4 import pandas as pd
```

### Question-1

Use a tuple to create 1,2,3 Dimension array and also check the dimension of array

In [51]:

```
1 # Creating a tuple Tup1 with integer elements
2 Tup1 = (10, 11, 12, 13)
3
4 # Creating a tuple Tup2 with tuples as elements
5 Tup2 = ((1, 2, 3, 4, 5), (6, 7, 8, 9, 10))
6
7 # Creating a nested tuple Tup3 with tuples and nested tuples as elements
8 Tup3 = (
9     ((15, 17, 16, 14, 13), (5, 6, 7, 10, 11)),
10    ((1, 14, 19, 5, 6), (6, 9, 11, 10, 15))
11 )
```

In [52]:

```
1 # Converting Tuples to a NumPy array
2
3 Arry1 = np.array(Tup1)
4 Arry2 = np.array(Tup2)
5 Arry3 = np.array(Tup3)
```

In [53]:

```
1 Arry1
```

Out[53]:

```
array([10, 11, 12, 13])
```

In [54]:

```
1 Arry2
```

Out[54]:

```
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10]])
```

In [55]:

```
1 Arry3
```

Out[55]:

```
array([[[15, 17, 16, 14, 13],
        [ 5,  6,  7, 10, 11]],

       [[ 1, 14, 19,  5,  6],
        [ 6,  9, 11, 10, 15]]])
```

In [56]:

```
1 # checking the dimensions of the all three arrays
2
3 print("First Array dimension: ",Arry1.ndim)
4 print("Second Array dimension: ",Arry2.ndim)
5 print("Third Array dimension: ",Arry3.ndim)
```

```
First Array dimension:  1
Second Array dimension: 2
Third Array dimension:  3
```

## Question-2

Access the 1st element created from 1d Array.

In [57]:

```
1 Arry1[0]
```

Out[57]:

```
10
```

## Question-3

Access the element on 1st row 2nd column by use of 2d array.

In [58]:

```
1 Arry2[0,1]
```

Out[58]:

2

## Question-4

---

Access the element on 2nd row 5th column by use of 2d array.

---

In [59]:

```
1 Arry2[1,4]
```

Out[59]:

10

## Question-5

---

Access the 3rd element of 2nd array of 1st array by using of 3d array.

---

In [60]:

```
1 Arry3
```

Out[60]:

```
array([[[15, 17, 16, 14, 13],
        [ 5,  6,  7, 10, 11]],
       [[ 1, 14, 19,  5,  6],
        [ 6,  9, 11, 10, 15]]])
```

In [61]:

```
1 Arry3[0,1,2]
```

Out[61]:

7

## Question-6

---

By use of negative indexing print the last element from 2 dimensional array

In [62]:

```
1 Array2
```

Out[62]:

```
array([[ 1,  2,  3,  4,  5],
       [ 6,  7,  8,  9, 10]])
```

In [63]:

```
1 Array2[-1,-1]
```

Out[63]:

```
10
```

## Question-7

Create one dim array and slice the elements from index 1 to 5 by creating 1D array

In [64]:

```
1 Array1
```

Out[64]:

```
array([10, 11, 12, 13])
```

In [65]:

```
1 # Slice elements from index 1 to index 5 (exclusive) from Array1
2 sliced_array1 = Array1[1:6]
3
4 # Print the sliced array sliced_array1
5 print(sliced_array1)
```

```
[11 12 13]
```

## Question-8

Slice the elements from index 4 to the end of the array.

In [66]:

```
1 # Define a tuple named Array_4 with given elements
2 Array_4 = (1, 2, 3, 6, 9, 10, 11)
3
4 # Slice elements from index 3 to the end of Array_4 and assign it to sliced1_array1
5 sliced1_array1 = Array_4[3:]
6
7 # Display the contents of sliced1_array1
8 sliced1_array1
```

Out[66]:

(6, 9, 10, 11)

## Question-9

---

By using concept of slicing return every other element from index 1 to 5 by creation of 1 D array.

---



In [67]:

```
1 # Slicing every other element from index 1 to 5
2 sliced_array = Array_4[1:6:2]
3
4 # Printing the sliced array
5 print(sliced_array)
```

(2, 6, 10)

## Question-10

---

Create 2D array, from 2nd element slice the elements from index 1 to 4.

---

In [68]:

```
1 import numpy as np
2
3 # Creating a 2D array using a nested list
4 array_2d = np.array([[1, 2, 3],
5                      [4, 5, 6],
6                      [7, 8, 9]])
7
8 # Displaying the 2D array
9 print("2D array: ",array_2d)
10
11 # Slicing elements from index 1 to index 4
12 sliced_array = array_2d[:, 1:4]
13
14 # Displaying the sliced array
15 for row in sliced_array:
16     for element in row:
17         print(element, end=" ")
18     print()
```

```
2D array:  [[1 2 3]
            [4 5 6]
            [7 8 9]]
2 3
5 6
8 9
```

## Question-11

Create 2D array, from 2nd element slice the elements from index 1 to 4.

In [69]:

```
1 a2 = np.arange(12,dtype=float).reshape(3,4)
```

In [70]:

```
1 a2[1,:]
```

Out[70]:

```
array([4., 5., 6., 7.])
```

## Question-12

Create a data frame by using a dictionary.

In [71]:

```
1 data = {  
2     'Name': ['Tina', 'Sam', 'Virat', 'Lisa'],  
3     'Age': [25, 28, 30, 22]  
4 }  
5  
6 df = pd.DataFrame(data)  
7 print(df)
```

|   | Name  | Age |
|---|-------|-----|
| 0 | Tina  | 25  |
| 1 | Sam   | 28  |
| 2 | Virat | 30  |
| 3 | Lisa  | 22  |

## Question-13

---

Check the shape of the data frame.

---

In [72]:

```
1 df.shape
```

Out[72]:

```
(4, 2)
```

## Question-14

---

add the new rows to dataframe.

---

In [73]:

```
1 fruits_df={"Fruit_Name":["Watermelon","Mango","Banana"],
2           "Price":[100,60,50],
3           "Quantity":[1,3,5]
4           }
5
6 Fruit1 = pd.DataFrame(fruits_df)
7 Fruit1
```

Out[73]:

|   | Fruit_Name | Price | Quantity |
|---|------------|-------|----------|
| 0 | Watermelon | 100   | 1        |
| 1 | Mango      | 60    | 3        |
| 2 | Banana     | 50    | 5        |

In [74]:

```
1 dict2={"Fruit_Name":["Grapes"], "Price":[90], "Quantity":[2]}
2
3 Fruit2 = pd.DataFrame(dict2)
4 Fruit2
```

Out[74]:

|   | Fruit_Name | Price | Quantity |
|---|------------|-------|----------|
| 0 | Grapes     | 90    | 2        |

In [75]:

```
1 Fruits =pd.concat([Fruit1, Fruit2], ignore_index=True)
2
3 Fruits
```

Out[75]:

|   | Fruit_Name | Price | Quantity |
|---|------------|-------|----------|
| 0 | Watermelon | 100   | 1        |
| 1 | Mango      | 60    | 3        |
| 2 | Banana     | 50    | 5        |
| 3 | Grapes     | 90    | 2        |

## Question-15

Create a line graph, bar graph and pie chart using matplotlib and also add the labels.



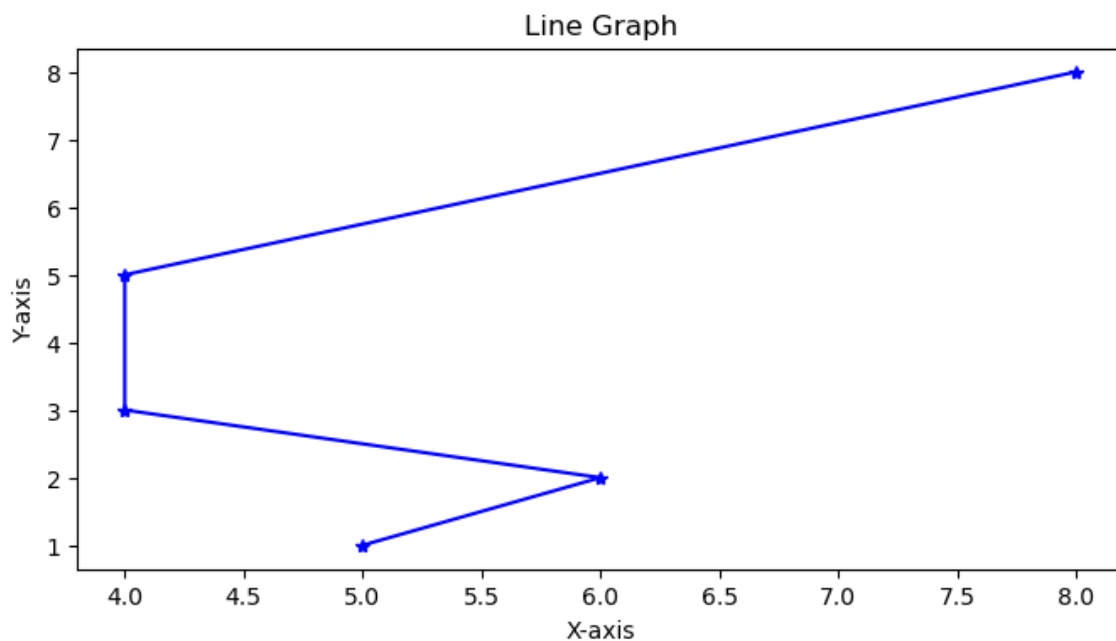
In [76]:

```
1 import matplotlib.pyplot as plt
```

### Line Graph

In [77]:

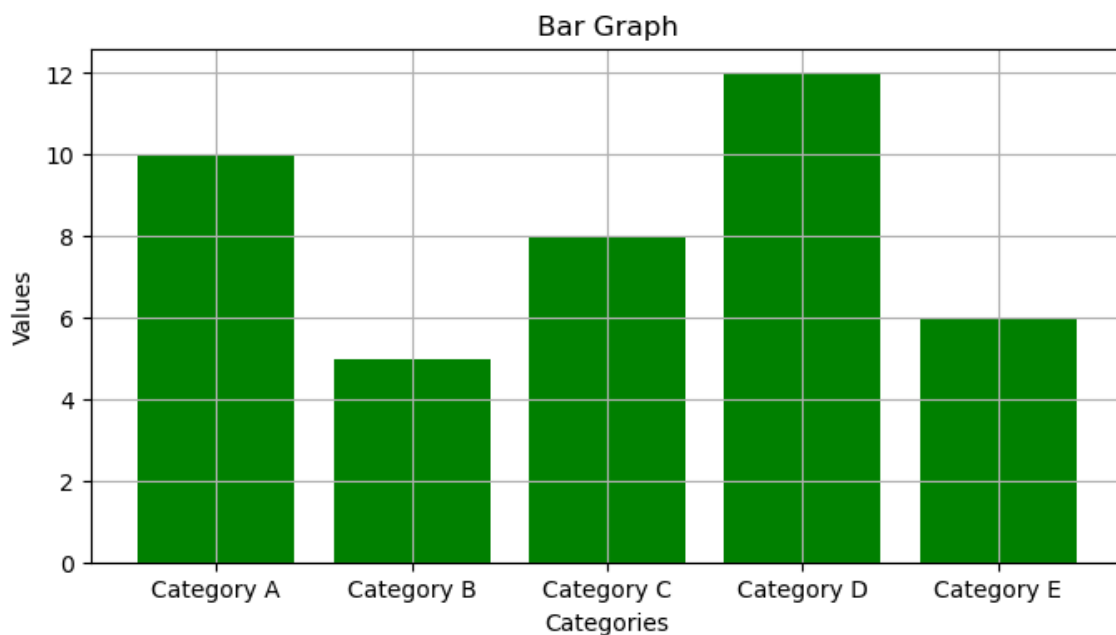
```
1 x_values = [5, 6, 4, 4, 8]
2 y_values = [1, 2, 3, 5, 8]
3
4 # Creating a line graph
5 plt.figure(figsize=(8, 4))
6 plt.plot(x_values, y_values, marker='*', linestyle='-', color='blue')
7 plt.xlabel('X-axis')
8 plt.ylabel('Y-axis')
9 plt.title('Line Graph')
10 plt.show()
```



### Bar Graph

In [78]:

```
1 categories = ['Category A', 'Category B', 'Category C', 'Category D', 'Category E']
2 y_values = [10, 5, 8, 12, 6]
3
4 # Creating a figure with a specific size
5 plt.figure(figsize=(8, 4))
6
7 # Creating a bar graph with the given categories and y-values
8 plt.bar(categories, y_values, color='green')
9
10 plt.xlabel('Categories') # Setting the label for the x-axis
11 plt.ylabel('Values') # Setting the label for the y-axis
12 plt.title('Bar Graph') # Setting the title for the graph
13
14 plt.grid(True) # Enabling the grid lines on the graph
15
16 plt.show() # Displaying the bar graph
```

**Pie chart**

In [79]:

```
1 # Define the sizes of the pie chart segments
2 pie_sizes = [30, 20, 15, 10, 25]
3
4 # Create a figure with a specific size
5 plt.figure(figsize=(6, 6))
6
7 # Create a pie chart with the given sizes and customization
8 plt.pie(pie_sizes, autopct='%1.1f%%', startangle=90,
9         colors=['red', 'blue', 'green', 'yellow', 'purple'])
10
11 plt.title('Pie Chart') # Set the title of the pie chart
12 plt.axis('equal') # Set the aspect ratio of the pie chart to be equal
13 plt.show() # Display the pie chart
```

Pie Chart

