Python NumPy Exercise

1: Create a 4X2 integer array and Prints its attributes

Note: The element must be a type of unsigned int16. And print the following Attributes: –

- The shape of an array.
- Array dimensions.
- The Length of each element of the array in bytes.

Expected Output:

```
Printing Array



[[64392 31655]

 [32579     0]

 [49248   462]

 [    0     0]]



Printing NumPy array Attributes



Array Shape is:  (4, 2)

Array dimensions are  2

Length of each element of array in bytes is  2
```

2: Create a 5X2 integer array from a range between 100 to 200 such that the difference between each element is 10

Expected Output:

```
Creating 5X2 array using numpy.arange

[[100 110]

 [120 130]

 [140 150]

 [160 170]

 [180 190]]
```

3: Following is the provided numPy array. Return array of items by taking the third column from all rows

```
sampleArray = numpy.array([[11 ,22, 33], [44, 55, 66], [77, 88, 99]])
```

Expected Output:

```
Printing Input Array

[[11 22 33]

 [44 55 66]

 [77 88 99]]



Printing array of items in the third column from all rows

[33 66 99]
```

4: Return array of odd rows and even columns from below numpy array

```
sampleArray = numpy.array([[3 ,6, 9, 12], [15 ,18, 21, 24],
[27 ,30, 33, 36], [39 ,42, 45, 48], [51 ,54, 57, 60]])
```

Expected Output:

```
Printing Input Array

[[ 3  6  9 12]

 [15 18 21 24]

 [27 30 33 36]

 [39 42 45 48]

 [51 54 57 60]]



Printing array of odd rows and even columns

[[ 6 12]

 [30 36]

 [54 60]]
```

5: Create a result array by adding the following two NumPy arrays. Next, modify the result array by calculating the square of each element

```
arrayOne = numpy.array([[5, 6, 9], [21 ,18, 27]])
arrayTwo = numpy.array([[15 ,33, 24], [4 ,7, 1]])
```

Expected Output:

```
addition of two arrays is
```

```
[[20 39 33]

 [25 25 28]]


Result array after calculating the square root of all elements


[[ 400 1521 1089]

 [ 625  625  784]]
```

6: Split the array into four equal-sized sub-arrays

Note: Create an 8X3 integer array from a range between 10 to 34 such that the difference between each element is 1 and then Split the array into four equal-sized sub-arrays.

Expected Output:

```
Creating 8X3 array using numpy.arange

[[10 11 12]

 [13 14 15]

 [16 17 18]

 [19 20 21]

 [22 23 24]

 [25 26 27]

 [28 29 30]

 [31 32 33]]
```

```
Dividing 8X3 array into 4 sub array


[array([[10, 11, 12],[13, 14, 15]]),

array([[16, 17, 18],[19, 20, 21]]),

array([[22, 23, 24],[25, 26, 27]]),

array([[28, 29, 30],[31, 32, 33]])]
```

7: Sort following NumPy array

- **Case 1**: Sort array by the second row
- **Case 2**: Sort the array by the second column

```
sampleArray = numpy.array([[34,43,73],[82,22,12],[53,94,66]])
```

Expected Output:

```
Printing Original array

[[34 43 73]

 [82 22 12]

 [53 94 66]]


Sorting Original array by second row

[[73 43 34]

 [12 22 82]

 [66 94 53]]


Sorting Original array by second column
```

```
[[82 22 12]

 [34 43 73]

 [53 94 66]]
```

8: Print max from axis 0 and min from axis 1 from the following 2-D array.

```
sampleArray = numpy.array([[34,43,73],[82,22,12],[53,94,66]])
```

Expected Output:

```
Printing Original array

[[34 43 73]

 [82 22 12]

 [53 94 66]]



Printing amin Of Axis 1

[34 12 53]



Printing amax Of Axis 0

[82 94 73]
```

9: Delete the second column from a given array and insert the following new column in its place.

```
sampleArray = numpy.array([[34,43,73],[82,22,12],[53,94,66]])
newColumn = numpy.array([[10,10,10]])
```

Expected Output:

```
Printing Original array

[[34 43 73]

 [82 22 12]

 [53 94 66]]



Array after deleting column 2 on axis 1

[[34 73]

 [82 12]

 [53 66]]



Array after inserting column 2 on axis 1

[[34 10 73]

 [82 10 12]

 [53 10 66]]
```

10: Create two 2-D arrays and Plot them using matplotlib