1) Create a 4X2 integer array and Prints its attributes Attributes like: a) The shape of an array. b) Array dimensions. c) The Length of each element of the array in bytes.

In [1]:

```
import numpy
```

In [2]:

```
firstArray = numpy.empty([4,2], dtype = numpy.uint16)
print("Printing Array")
print(firstArray)

print("Printing numpy array Attributes")
print("1> Array Shape is: ", firstArray.shape)
print("2>. Array dimensions are ", firstArray.ndim)
print("3>. Length of each element of array in bytes is ", firstArray.itemsize)
```

```
Printing Array
[[36887 27614]
 [36039 55316]
 [29912 23646]
 [25615 63143]]
Printing numpy array Attributes
1> Array Shape is:  (4, 2)
2>. Array dimensions are  2
3>. 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

In [3]:

```
import numpy

print("Creating 5X2 array using numpy.arange")
sampleArray = numpy.arange(100, 200, 10)
sampleArray = sampleArray.reshape(5,2)
print (sampleArray)
```

```
Creating 5X2 array using numpy.arange
[[100 110]
 [120 130]
 [140 150]
 [160 170]
 [180 190]]
```

3) Add the following two NumPy arrays import

numpy

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

In [4]:

```
import numpy

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

resultArray  = arrayOne + arrayTwo
print("addition of two arrays is \n")
print(resultArray)
```

```
addition of two arrays is

[[20 39 33]
 [25 25 28]]
```

4) 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]])]

In [5]:

```
import numpy

print("Creating 8X3 array using numpy.arange")
sampleArray = numpy.arange(10, 34, 1)
sampleArray = sampleArray.reshape(8,3)
print (sampleArray)

print("\nDividing 8X3 array into 4 sub array\n")
subArrays = numpy.split(sampleArray, 4)
print(subArrays)
```

```
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]])]
```

5) Write a NumPy program to generate six random integers between 10 and 30.

Expected Output: [20 28 27 17 28 29]

In [6]:

```
import numpy as np
a = np.random.randint(low=10, high=30, size=6)
print(a)
```

```
[16 15 23 29 25 23]
```

6) Write a NumPy program to generate five random numbers from the normal distribution. Expected Output: [-0.43262625 -1.10836787 1.80791413 0.69287463 -0.53742101]

In [8]:

```
import numpy as np
b = np.random.normal(size=5)
print(b)
```

```
[-0.80976728 -1.04908966  1.52767249  1.73141854 -0.73176291]
```

7) Write a NumPy program to create a 3x3x3 array with random values.

In [9]:

```
import numpy as np
x = np.random.random((3,3,3))
print(x)
```

```
[[[0.66229147 0.517888   0.83716288]
  [0.43823503 0.64184822 0.13786912]
  [0.6831581  0.65551901 0.0828665 ]]

 [[0.07725439 0.77181578 0.58096607]
  [0.75145154 0.81758833 0.18284788]
  [0.12277694 0.2063462  0.62442645]]

 [[0.9520522  0.82578813 0.37361802]
  [0.44415853 0.62382735 0.63497914]
  [0.32032552 0.50885417 0.14222225]]]
```

8) Write a NumPy program to normalize a 3x3 random matrix.

In [10]:

```
import numpy as np
x= np.random.random((3,3))
print("Original Array:")
print(x)
xmax, xmin = x.max(), x.min()
x = (x - xmin)/(xmax - xmin)
print("After normalization:")
print(x)
```

```
Original Array:
[[0.26008764 0.51994033 0.03344339]
 [0.74146406 0.05411541 0.57326899]
 [0.51291129 0.67577377 0.33652798]]
After normalization:
[[0.32010965 0.68712251 0.        ]
 [1.         0.02919692 0.76244327]
 [0.67719477 0.90721982 0.42807308]]
```

9) Write a NumPy program to shuffle numbers between 0 and 10 (inclusive).

In [11]:

```
import numpy as np
x = np.arange(10)
np.random.shuffle(x)
print(x)
print("Same result using permutation():")
print(np.random.permutation(10))
```

```
[2 9 0 8 1 7 6 5 3 4]
Same result using permutation():
[5 4 6 3 1 8 0 9 7 2]
```

10) Write a NumPy program to create a 5x5 array with random values and find the minimum and maximum values

In [12]:

```
import numpy as np
x = np.random.random((5,5))
print("Original Array:")
print(x)
xmin, xmax = x.min(), x.max()
print("Minimum and Maximum Values:")
print(xmin, xmax)
```

```
Original Array:
[[0.10918321 0.33442558 0.8739067  0.53965127 0.86052153]
 [0.52501156 0.01555026 0.50925839 0.09406969 0.97434145]
 [0.19492055 0.57257297 0.73726182 0.98656428 0.51929817]
 [0.69082862 0.15061504 0.11233091 0.6942625  0.71285474]
 [0.1598169  0.20933851 0.81955428 0.4526172  0.17884658]]
Minimum and Maximum Values:
0.015550264962885918 0.9865642822741096
```