```
import numpy as np
np.__version__
```

```
'1.19.5'
```

1. Write a NumPy program to convert a given array into a list and then convert it into array again.

```
import numpy as np
A=np.arange(0,10)
print(type(A))
A=A.tolist()
print(type(A))
A=np.array(A)
print(type(A))
```

```
<class 'numpy.ndarray'>
<class 'list'>
<class 'numpy.ndarray'>
```

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

```
import numpy as np
A=np.arange(100,200,10).reshape(5,2)
print(A)
```

```
[[100 110]
 [120 130]
 [140 150]
 [160 170]
 [180 190]]
```

3. Add the two 2D NumPy arrays and modify the resulting array by calculating the square root of each element

```
import numpy as np
A=np.arange(10,20).reshape(5,2)
B=np.arange(15,25).reshape(5,2)
C=A+B
C=np.sqrt(C)
print(C)
```

```
     [[5.         5.19615242]
      [5.38516481 5.56776436]
      [5.74456265 5.91607978]
      [6.08276253 6.244998  ]
      [6.40312424 6.55743852]]
```

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.

```
import numpy as np
A=np.arange(10,34).reshape(8,3)
A=np.split(A,4)
print(A)
```

```
     [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. Consider the following array: sampleArray = ([[34 43 73] [82 22 12] [53 94 66]])

i. Sort above array by second row

```
import numpy as np
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
print(sampleArray)
sorted_array = sampleArray [ :, sampleArray[1].argsort()]
print(sorted_array)
```

```
[[34 43 73]
 [82 22 12]
 [53 94 66]]
[[73 43 34]
 [12 22 82]
 [66 94 53]]
[[82 22 12]
 [34 43 73]
 [53 94 66]]
```

ii. Sort above array by second column

```
import numpy as np
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
print(sampleArray)
sorted_array = sampleArray[np.argsort(sampleArray[:, 1])]
print(sorted_array)
```

```
[[34 43 73]
 [82 22 12]
 [53 94 66]]
[[82 22 12]
 [34 43 73]
 [53 94 66]]
```

iii. Print max from axis 0 and min from axis 1

```
import numpy as np
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
print(sampleArray)
print(sampleArray.max(axis=0))
print(sampleArray.min(axis=1))
```

```
[[34 43 73]
 [82 22 12]
 [53 94 66]]
[82 94 73]
[34 12 53]
```

iv. Delete col 2 and insert new Column numpy.array([[10,10,10]]) in its place

```
import numpy as np
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
sampleArray=np.delete(sampleArray,obj=1,axis=1)
sampleArray = np.insert(sampleArray, 1, [[10, 10, 10]],axis=1)
print(sampleArray)
```

```
[[34 10 73]
 [82 10 12]
 [53 10 66]]
```

6. Remove all the elements from an array that exist in another array.

```
import numpy as np
A=np.arange(10,20)
B=np.arange(1,15)
#A=[i for i in A if i not in B]
np.setdiff1d(A,B)
print(A)
```

```
[10 11 12 13 14 15 16 17 18 19]
```

7. Swap two columns in a 2d NumPy array.

```
import numpy as np
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
#sampleArray[:,0],sampleArray[:,1] = sampleArray[:,1],sampleArray[:,0
sampleArray[:,[0, 1]] = sampleArray[:,[1, 0]]
print(sampleArray)
```

```
[[43 34 73]
 [22 82 12]
 [94 53 66]]
```

8. Swap two rows in a 2d NumPy array.

```
import numpy as np
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
sampleArray[[0, 1],:] = sampleArray[[1, 0],:]
print(sampleArray)
```

```
[[82 22 12]
 [34 43 73]
 [53 94 66]]
```

9. Reverse the order of rows of a 2D array

```
import numpy as np
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
#sampleArray[:,:] = sampleArray[::-1,:]
sampleArray[:] = sampleArray[::-1]
print(sampleArray)
```

```
     [[53 94 66]
      [82 22 12]
      [34 43 73]]
```

10. Reverse the order of columns of a 2D array

```
import numpy as np
sampleArray = np.array([[34,43,73],[82,22,12],[53,94,66]])
sampleArray[:,:] = sampleArray[:,::-1]
print(sampleArray)
```

```
     [[73 43 34]
      [12 22 82]
      [66 94 53]]
```

11. Retrieve common items between two python NumPy arrays?

```
import numpy as np
A=np.arange(10,20)
B=np.arange(1,15)
print(np.intersect1d(A, B))
```

```
     [10 11 12 13 14]
```

12. Retrive indices where elements of two arrays match

```
import numpy as np
A=np.array([1,2,3,4,5,6,7,8])
B=np.array([1,1,2,3,5,6,9,8])

print(np.where(A == B))
```

```
    (array([0, 4, 5, 7]),)
```

13. Get all items between 5 and 10 from an array.

```
import numpy as np
A=np.arange(1,20)

print(np.where((A>5) & (A<10)))
#print(np.where(np.logical_and(A>5, A<10)))
```

```
    (array([5, 6, 7, 8]),)
```

14. For a 1D array with numeric values, find minimum, maximum, mean, median, standard deviation, 5th and 95th percentile, unique values, count of unique values, and the most frequent value.

```
import numpy as np
import statistics
A=np.arange(10,20)
A = np.concatenate((A, np.arange(1,15),[10]))
A=np.sort(A)
print(np.amin(A))
print(np.amax(A))
print(np.mean(A))
print(np.median(A))
print(np.std(A))
print(np.percentile(A,5))
print(np.percentile(A,95))
print(np.unique(A))
print(len(np.unique(A)))
print(statistics.mode(A))
```

```
    1
    19
```

```
10.4
11.0
4.882622246293481
2.2
17.799999999999997
[ 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19]
19
10
```

15. Write a NumPy program to create a 10x10 matrix, in which all the elements on the borders should be equal to 1, and rest others should be 0.

```
import numpy as np
A = np.ones((10, 10))
A[1:-1, 1:-1] = 0
print(x)
```

```
[[1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [1. 0. 0. 0. 0. 0. 0. 0. 0. 1.]
 [1. 1. 1. 1. 1. 1. 1. 1. 1. 1.]]
```

16. Write a NumPy program to create a 5x5 zero matrix with elements on the main diagonal equal to 1, 2, 3, 4, 5

```
import numpy as np
#A = np.eye(5)
A=np.diagflat([1,2,3,4,5])
```

```
print(A)

    [[1 0 0 0 0]
     [0 2 0 0 0]
     [0 0 3 0 0]
     [0 0 0 4 0]
     [0 0 0 0 5]]
```

```
import numpy as np
A=np.arange(10,20)
A = np.concatenate((A, np.arange(1,15),[10]))
print(np.count_nonzero(A>10))
A[np.nonzero(A>10)]=A[np.nonzero(A>10)]*10
print(A)

    13
    [ 10 110 120 130 140 150 160 170 180 190   1   2   3   4   5
       9  10 110 120 130 140  10]
```