ASSIGNMENT NO :- 1

1)Create a 4X2 float array and Prints its attributes

```
import numpy as np

a= np.array([[1.5, 2.3], [3.1, 4.2], [5.7, 6.8], [7.9, 8.0]])

print("Specific Array:\n",a)

print("Shape:",a.shape)

print("Size:",a.size)

print("Datatype:",a.dtype)
```

OUTPUT:-

```
Specific Array:

 [[1.5 2.3]

 [3.1 4.2]

 [5.7 6.8]

 [7.9 8. ]]

Shape: (4, 2)

Size: 8

Datatype: float64
```

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

```
import numpy as np

a=np.arange(100,1000,200)

r=a.reshape(5,2)

print(r)
```

OUTPUT:-

```
ValueError                          Traceback (most recent call last)

Cell In[157], line 3

      1 import numpy as np

      2 a=np.arange(100,1000,200)

----> 3 r=a.reshape(5,2)

      4 print(r)


ValueError: cannot reshape array of size 5 into shape (5,2)
```

3) Print max from axis 0 and min from axis 1 from the following 3-D array.

```
import numpy as np

a=np.array([[[1, 2, 3], [13, 14, 15], [16, 17, 18]],

[[20, 21, 22], [23, 24, 25], [26, 27, 28]],

[[30, 31, 32], [33, 34, 35], [36, 37, 38]]])

print("Original 3D Array:")

print(a)
print("max from axis-0:\n",  np.max(a, axis=0))
print("min from axis-1:\n",  np.min(a, axis=1))
```

OUTPUT:-

```
Original 3D Array:
[[[ 1  2  3]
  [13 14 15]
  [16 17 18]]


 [[20 21 22]
  [23 24 25]
  [26 27 28]]


 [[30 31 32]
  [33 34 35]
  [36 37 38]]]
max from axis-0:
 [[30 31 32]
 [33 34 35]
 [36 37 38]]
min from axis-1:
 [[ 1  2  3]
 [20 21 22]
 [30 31 32]]
```

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)
sub_arrays = np.split(a, 4)
print("Original Array:")
print(a)
for s in sub_arrays:
    print("\nSub-array:")
    print(s)
```

OUTPUT:-

```
Original Array:
[[10 11 12]
 [13 14 15]
 [16 17 18]
 [19 20 21]
 [22 23 24]
 [25 26 27]
 [28 29 30]
 [31 32 33]]

Sub-array:
```

```
[[10 11 12]
 [13 14 15]]

Sub-array:
[[16 17 18]
 [19 20 21]]

Sub-array:
[[22 23 24]
 [25 26 27]]

Sub-array:
[[28 29 30]
 [31 32 33]]
```

5) Return array of odd rows and even columns from 2D numpy array

```
import numpy as np
a= np.array([[1, 2, 3, 4],
          [5, 6, 7, 8],
          [9, 10, 11, 12],
          [13, 14, 15, 16]])
result = a[::2, 1::2]
print("Odd Row & Even Column of 2-D Array:")
print(result)
```

OUTPUT:-

```
Odd Row & Even Column of 2-D Array:
[[ 2  4]
 [10 12]]
```

6) Create a Numpy array filled with all zeros and ones

```
import numpy as np
a=np.zeros((3, 3),dtype='int')
b=np.ones((3, 3),dtype='int')
print("All Zeros Array:\n",a)
print("All Ones Array:\n",b)
```

OUTPUT:-

```
All Zeros Array:
 [[0 0 0]
 [0 0 0]
 [0 0 0]]
All Ones Array:
 [[1 1 1]
 [1 1 1]
 [1 1 1]]
```

7) Remove rows and columns in Numpy array that contains non-numeric values.

```
import numpy as np
a=np.array([[1, 2, 'nan'], [3, 4, 5]])
a=np.where(a=='nan',np.nan,a)
a=a.astype(float)
a=a[~np.isnan(a).any(axis=1)]
print("\nFiltered array:\n",a)
```

OUTPUT:-

```
Filtered array:
 [[3. 4. 5.]]
```

8) Write a program to compare two NumPy arrays and return common items between two numpy arrays?

```
import numpy as np
a1 = np.array([1, 2, 3, 4, 5])
a2 = np.array([4, 5, 6, 7, 8])
c=np.intersect1d(a1,a2)
print("Common items:",c)
```

OUTPUT:-

```
Common items: [4 5]
```

9) To get all 2D diagonals of a 3D NumPy array

```
a= np.array([[[1, 2, 3],
               [4, 5, 6],
               [7, 8, 9]],

              [[10, 11, 12],
               [13, 14, 15],
               [16, 17, 18]],

              [[19, 20, 21],
               [22, 23, 24],
               [25, 26, 27]]])
print("\n2D Diagonals of 3D array:\n",np.diagonal(a,axis1=1,axis2=2))
```

OUTPUT:-

```
2D Diagonals of 3D array:
 [[ 1  5  9]
 [10 14 18]
 [19 23 27]]
```

10) To sort and reverse a numpy array

```
import numpy as np
a=np.array([3, 1, 4, 1, 5, 9, 2])
sort_a=np.sort(a)
rev_a=sort_a[::-1]
print("Sorted Array:\n",sort_a)
print("Reversed Array:\n",rev_a)
```

OUTPUT:-

```
Sorted Array:
 [1 1 2 3 4 5 9]
Reversed Array:
 [9 5 4 3 2 1 1]
```

11)Write a program to add row/columns in numpy array

```
import numpy as np
a=np.array([[1, 2, 3],
            [4, 5, 6]])
print("Original array:")
print(a)

r=np.array([7, 8, 9])
new_r= np.vstack((a,r))
c= np.array([[10], [11], [12]])
```

```
new_c=np.hstack((new_r,c))
print("\nArray after adding a new row:")
print(new_r)
print("\nArray after adding a new column:")
print(new_c)
```

OUTPUT:-

```
Original array:
[[1 2 3]
 [4 5 6]]

Array after adding a new row:
[[1 2 3]
 [4 5 6]
 [7 8 9]]

Array after adding a new column:
[[ 1  2  3 10]
 [ 4  5  6 11]
 [ 7  8  9 12]]
```

12)Find the unique and the count of unique values.

```
import numpy as np
a=np.array([1, 2, 2, 3, 4, 4, 4, 5, 6])
```

```
uni_val, cnt= np.unique(a, return_counts=True)
print("Unique values:", uni_val)
print("Counts of unique values:", cnt)
```

OUTPUT:-

```
Unique values: [1 2 3 4 5 6]
Counts of unique values: [1 2 1 3 1 1]
```

13)Replace all elements of array by ‘Nan’ which are less than 10.

```
import numpy as np
a=np.array([5, 15, 3, 20, 8, 12, 7], dtype='float')
a[a<10]=np.nan
print("Array after replacing elements less than 10 with NaN:")
print(a)
```

OUTPUT:-

```
Array after replacing elements less than 10 with NaN:
[nan 15. nan 20. nan 12. nan]
```

14)Extract the first four columns of this 2-D array

```
import numpy as np
a=np.array([[1, 2, 3, 4, 5],
          [6, 7, 8, 9, 10],
          [11, 12, 13, 14, 15]])
four_col=a[:, :4]
print("First four columns:")
print(four_col)
```

OUTPUT:-

```
First four columns:
[[ 1  2  3  4]
 [ 6  7  8  9]
 [11 12 13 14]]
```

15)To compute the mean, median, standard deviation of a numpy array

```
import numpy as np
a=np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
print("Mean Of Array:",np.mean(a))
print("Median Of Array:",np.median(a))
print("Standard Deviation Of Array:",np.std(a))
```

OUTPUT:-

```
Mean Of Array: 5.5
Median Of Array: 5.5
```

```
Standard Deviation Of Array: 2.8722813232690143
```

16)Finding the k smallest values of a NumPy array

```
import numpy as np
a=np.array([7, 2, 3, 1, 6, 5, 8, 4])
k=3
s=np.partition(a,k)[:k]
print("The k smallest values:",s)
```

OUTPUT:-

```
The k smallest values: [2 1 3]
```

17)To get the n-largest values of an array using NumPy.

```
import numpy as np
a=np.array([7, 2, 3, 1, 6, 5, 8, 4])
n=3
large= np.partition(a,-n)[-n:]
print("The n largest values:",large)
```

OUTPUT:-

```
The n largest values: [6 8 7]
```

18)To normalize an 2D array and the values range exactly between 0 and 1

```
import numpy as np
a=np.array([[1, 2, 3],
            [4, 5, 6],
            [7, 8, 9]])
min=a.min()
max=a.max()
normalized_arr=(a- min)/(max- min)
print("Normalized array:\n",normalized_arr)
```

OUTPUT:-

```
Normalized array:
 [[0.    0.125 0.25 ]
 [0.375 0.5   0.625]
 [0.75  0.875 1.   ]]
```

19)Create a Numpy array with random values

```
import numpy as np
rarr = np.random.rand(3, 4)
```

```
print("Random array:\n",rarr)
```

OUTPUT:-

```
Random array:
 [[0.87361892 0.09185488 0.92153211 0.85057793]
 [0.404072   0.20578642 0.62580829 0.51940987]
 [0.67310765 0.30238585 0.45038569 0.50686667]]
```

20)Insert a space between characters of all the elements of a given NumPy array.

```
import numpy as np
a=np.array(['hello', 'world', 'numpy'])
sarr=np.char.join(' ', a)
print("Array with spaces between characters:\n",sarr)
```

OUTPUT:-

```
Array with spaces between characters:
 ['h e l l o' 'w o r l d' 'n u m p y']
```

21)Find the length of each string element in the Numpy array

```
import numpy as np
```

```
a=np.array(['hello', 'world', 'numpy'])
l=np.char.str_len(a)
print("Lengths of each string element:\n",l)
```

OUTPUT:-

```
Lengths of each string element:
 [5 5 5]
```

22)Change the case to uppercase or lowercase of elements of an array

```
import numpy as np
a=np.array(['Hello', 'World', 'NUMpy'])
u=np.char.upper(a)
l=np.char.lower(a)
print("Array in uppercase:",u)
print("Array in lowercase:",l)
```

OUTPUT:-

```
Array in uppercase: ['HELLO' 'WORLD' 'NUMPY']
Array in lowercase: ['hello' 'world' 'numpy']
```