NumPy Practical 1

Roll no 407

name :- Nayan khuje

In [1]:
```
import numpy as np
```

Create the 1-dimensional array.

a) Name continent, consist of name of all continents. (Q3 included)

In [2]:
```
Continent=np.array(['Asia','africa','Europe','North America','South America','Austalia/Oceania','antarctica'])
print(Continent)
print(Continent.ndim)
print(Continent.shape)
print(Continent.dtype)
print(Continent.itemsize)
```

```
['Asia' 'africa' 'Europe' 'North America' 'South America'
 'Austalia/Oceania' 'antarctica']
1
(7,)
<U16
64
```

b) Named ocean, consists of name of all ocean. (Q3 included)

In [3]:
```
ocean=np.array(['Pacific Ocean','Atlantic Ocean','Indian Ocean','Antarctic Ocean (Southem Ocean)','Arctic Ocean'])
print(ocean)
print(ocean.ndim)
print(ocean.shape)
print(ocean.dtype)
print(ocean.itemsize)
```

```
['Pacific Ocean' 'Atlantic Ocean' 'Indian Ocean'
 'Antarctic Ocean (Southem Ocean)' 'Arctic Ocean']
1
(5,)
<U31
124
```

c) Named river, consists of seven prominent rivers of world. (Q3 included)

In [4]:
```
river=np.array(['The Nile (North Africa)','The Amazon (South America)'
,'The Yangtze (China)',
'The Congo (Central Africa)',
'The Mississippi (North America)','The Volga (Russia)',
'The Danube (Central and Eastern Europe)'])
print(river)
print(river.ndim)
print(river.shape)
print(river.dtype)
print(river.itemsize)
```

```
['The Nile (North Africa)' 'The Amazon (South America)'
 'The Yangtze (China)' 'The Congo (Central Africa)'
 'The Mississippi (North America)' 'The Volga (Russia)'
 'The Danube (Central and Eastern Europe)']
1
(7,)
<U39
156
```

d) Named prime consists of first 10 prime numbers between 100 and 200. (Q3 included)

In [5]:
```
prime=np.array([101,103,107,109,113,127,131,137,139,149])
print(prime)
print(prime.ndim)
print(prime.shape)
print(prime.dtype)
print(prime.itemsize)
```

```
[101 103 107 109 113 127 131 137 139 149]
1
(10,)
int32
4
```

e) Named planet, consists of name of plant in the universe of earth. (Q3 included)

In [6]:
```
planet=np.array(['Mercury' , 'Venus ' , ' Earth' , 'Mars ' , 'Jupiter ','Saturn' , 'Uranus '
'Neptune' , 'Pluto'])
print(planet)
```

```
print(planet.ndim)
print(planet.shape)
print(planet.dtype)
print(planet.itemsize)
```

```
['Mercury' 'Venus ' ' Earth' 'Mars ' 'Jupiter ' 'Saturn' 'Uranus Neptune'
 'Pluto']
1
(8,)
<U14
56
```

f) Named Fibonacci, consists of first 10 Fibonacci number divided by 5. (Q3 included)

In [7]:
```
fibonacci_div_5=np.array([0, 1/5, 1/5, 2/5, 3/5, 5/5, 8/5, 13/5, 21/5, 34/5])
print(fibonacci_div_5)
print(fibonacci_div_5.ndim)
print(fibonacci_div_5.shape)
print(fibonacci_div_5.dtype)
print(fibonacci_div_5.itemsize)
```

```
[0.  0.2 0.2 0.4 0.6 1.  1.6 2.6 4.2 6.8]
1
(10,)
float64
8
```

g) Named floats, consists of any seven numbers out of which 5 are float numbers and remaining two are integers

In [8]:
```
floats=np.array([120.12,42.598,48.46,895.124,45.897,4,9])
print(floats)
print(floats.ndim)
print(floats.shape)
print(floats.dtype)
print(floats.itemsize)
```

```
[120.12   42.598  48.46  895.124  45.897   4.      9.   ]
1
(7,)
float64
8
```

h) Named mixed, consists of 2 string values, 2 float numbers and 2 integers. (Q3 included)

In [9]:
```
mixed=np.array(['nayan','nano',23.5687,45.78,8,9])
print(mixed)
print(mixed.ndim)
print(mixed.shape)
print(mixed.dtype)
print(mixed.itemsize)
```

```
['nayan' 'nano' '23.5687' '45.78' '8' '9']
1
(6,)
<U32
128
```

i) Named arang, with first value 4, last value less than 100, in the step of 3. (Q3 included)

In [10]:
```
aranging=np.arange(4,100,3)
print(aranging)
print(aranging.ndim)
print(aranging.shape)
print(aranging.dtype)
print(aranging.itemsize)
```

```
[ 4  7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73
 76 79 82 85 88 91 94 97]
1
(32,)
int32
4
```

Q2. Create 2 – dimensional array for the following and print them: (Q3 included)

a) Named seven, with two rows with first row consists of seven sisters of India and second row consists of their capitals. (Q3 included)

In [11]:
```
seven=np.array([['Arunachal Pradesh','Assam','Nagaland','Manipur','Meghalaya','Tripura','Mizoram'],['Itanagar','Dispu
print(seven)
print(seven.ndim)
print(seven.shape)
print(seven.dtype)
print(seven.itemsize)
```

```
[['Arunachal Pradesh' 'Assam' 'Nagaland' 'Manipur' 'Meghalaya' 'Tripura'
  'Mizoram']
 ['Itanagar' 'Dispur' 'Kohima' 'Imphal' 'Shilong' 'Agartala' 'Aizawl']]
2
(2, 7)
<U17
68
```

b) Named null, with two rows and three columns with all entries zero. (Q3 included)

In [12]:
```
null=np.array([[0,0,0],[0,0,0]])
print(null)
print(null.ndim)
print(null.shape)
print(null.dtype)
print(null.itemsize)
```

```
[[0 0 0]
 [0 0 0]]
2
(2, 3)
int32
4
```

In [13]:
```
null2=np.zeros([2,3])
print(null2)
print(null2.ndim)
print(null2.shape)
print(null2.dtype)
print(null2.itemsize)
```

```
[[0. 0. 0.]
 [0. 0. 0.]]
2
(2, 3)
float64
8
```

c) Named allones, with 4 rows and 5 columns with all entries 1. (Q3 included)

In [14]:
```
allones=np.ones([4,5])
print(allones)
print(allones.ndim)
print(allones.shape)
print(allones.dtype)
print(allones.itemsize)
```

```
[[1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]
 [1. 1. 1. 1. 1.]]
2
(4, 5)
float64
8
```

d) Named pi, with 3 rows and 3 columns with all entries approximate value of pi up to 5 decimal places. (Q3 included)

In [15]:
```
pi=np.array([[3.14159,3.14159,3.14159],[3.14159,3.14159,3.14159],[3.14159,3.14159,3.14159]])
print(pi)
print(pi.ndim)
print(pi.shape)
print(pi.dtype)
print(pi.itemsize)
```

```
[[3.14159 3.14159 3.14159]
 [3.14159 3.14159 3.14159]
 [3.14159 3.14159 3.14159]]
2
(3, 3)
float64
8
```

e) Named data with 2 rows, where one row is name of 5 person and second name is its age. (Q3 included)

In [16]:
```
data=np.array([['nayan','nano','prachiti','mansi','shalu'],[20,58,69,45,100]])
print(data)
print(data.ndim)
print(data.shape)
print(data.dtype)
print(data.itemsize)
```

```
[['nayan' 'nano' 'prachiti' 'mansi' 'shalu']
 ['20' '58' '69' '45' '100']]
2
(2, 5)
<U11
44
```

Q3. Display the dimension, shape, size, data type and item size of all the

ndarray created in 1 and 2,

Q4. Display first, second, last and second last elements of all the array created in above examples.

for Q1

In [17]:
```
#for a
print(Continent[0]) #first
print(Continent[1]) #second
print(Continent[-1]) #last
print(Continent[-2]) #second last
```

```
Asia
africa
antarctica
Austalia/Oceania
```

In [18]:
```
#for b
print(ocean[0]) #first
print(ocean[1]) #second
print(ocean[-1]) #last
print(ocean[-2]) #second last
```

```
Pacific Ocean
Atlantic Ocean
Arctic Ocean
Antarctic Ocean (Southem Ocean)
```

In [19]:
```
#for c
print(river[0]) #first
print(river[1]) #second
print(river[-1]) #last
print(river[-2]) #second last
```

```
The Nile (North Africa)
The Amazon (South America)
The Danube (Central and Eastern Europe)
The Volga (Russia)
```

In [20]:
```
#for d
print(prime[0]) #first
print(prime[1]) #second
print(prime[-1]) #last
print(prime[-2]) #second last
```

```
101
103
149
139
```

In [21]:
```
#for e
print(planet[0]) #first
print(planet[1]) #second
print(planet[-1]) #last
print(planet[-2]) #second last
```

```
Mercury
Venus
Pluto
Uranus Neptune
```

In [22]:
```
#for f
print(fibonacci_div_5[0]) #first
print(fibonacci_div_5[1]) #second
print(fibonacci_div_5[-1]) #last
print(fibonacci_div_5[-2]) #second last
```

```
0.0
0.2
6.8
4.2
```

In [23]:
```
#for g
print(floats[0]) #first
print(floats[1]) #second
print(floats[-1]) #last
print(floats[-2]) #second last
```

```
120.12
42.598
9.0
4.0
```

In [24]:
```
#for h
print(mixed[0]) #first
print(mixed[1]) #second
print(mixed[-1]) #last
print(mixed[-2]) #second last
```

```
nayan
nano
9
8
```

In [25]:
```
#for i
print(aranging[0]) #first
print(aranging[1]) #second
print(aranging[-1]) #last
print(aranging[-2]) #second last
```

```
4
7
97
94
```

Q5. Display all the elements from 2 to second last in all the array created in question 1,

In [26]:
```
#for a
print(Continent[1:-1])
```

```
['africa' 'Europe' 'North America' 'South America' 'Austalia/Oceania']
```

In [27]:
```
#for b
print(ocean[1:-1])
```

```
['Atlantic Ocean' 'Indian Ocean' 'Antarctic Ocean (Southem Ocean)']
```

In [28]:
```
#for c
print(river[1:-1])
```

```
['The Amazon (South America)' 'The Yangtze (China)'
 'The Congo (Central Africa)' 'The Mississippi (North America)'
 'The Volga (Russia)']
```

In [29]:
```
#for d
print(prime[1:-1])
```

```
[103 107 109 113 127 131 137 139]
```

In [30]:
```
#for e
print(planet[1:-1])
```

```
['Venus ' ' Earth' 'Mars ' 'Jupiter ' 'Saturn' 'Uranus Neptune']
```

In [31]:
```
#for f
print(fibonacci_div_5[1:-1])
```

```
[0.2 0.2 0.4 0.6 1.  1.6 2.6 4.2]
```

In [32]:
```
#for g
print(floats[1:-1])
```

```
[ 42.598  48.46  895.124  45.897   4.   ]
```

In [33]:
```
#for h
print(mixed[1:-1])
```

```
['nano' '23.5687' '45.78' '8']
```

In [34]:
```
#for i
print(aranging[1:-1])
```

```
[ 7 10 13 16 19 22 25 28 31 34 37 40 43 46 49 52 55 58 61 64 67 70 73 76
 79 82 85 88 91 94]
```

Q6. Display the elements from second row, third column from the examples created in question 2.

In [35]:
```
#for a
print(seven[1,2])
```

```
Kohima
```

In [36]:
```
#for b
print(null[1,2])
```

```
0
```

In [37]:
```
#for c
print(allones[1,2])
```

```
1.0
```

In [38]:
```
#for d
print(pi[1,2])
```

```
3.14159
```

In [39]:
```
#for e
print(data[1,2])
```

In [ ]:

In [ ]: