```
# Working on Google Colab
import numpy as np
import pandas as pd


# Question 1-A
import math
x = 5.5
print("Floor value    : ",math.floor(x));
print("Ceil Value     : ",math.ceil(x));
print("Square value   : ",pow(x,2));
print("Sqrt Value     : ",math.sqrt(x));
print("Int Sqrt Value : ",math.isqrt(int(x)));
```

```
Floor value    :  5
Ceil Value     :  6
Square value   :  30.25
Sqrt Value     :  2.345207879911715
Int Sqrt Value :  2
```

```
# Question 1-B
arr = np.random.rand(3,3)
print("Numpy Array\n",arr)
print("Type of Array  : ",type(arr))
print("Dimension      : ", arr.ndim)
print("Shape          : ",arr.shape)
print("Total elements : ",arr.size)
print("Sum of array   : ",arr.sum())
print("Sum of array   : ",np.sum(arr))
print("Mean of array  : ",arr.mean())
print("Mean of array  : ",np.mean(arr))
print("Sorted array   : ",arr.sort())
print("Sorted array   : ",np.sort(arr))
print("Sin of array   : ",np.sin(arr))
```

```
Numpy Array
 [[0.81751272 0.60293683 0.44297591]
 [0.53680069 0.43862026 0.0450431 ]
 [0.24909415 0.90712041 0.29692352]]
Type of Array  :  <class 'numpy.ndarray'>
Dimension      :  2
Shape          :  (3, 3)
Total elements :  9
Sum of array   :  4.337027584137034
Sum of array   :  4.337027584137034
Mean of array  :  0.48189195379300376
Mean of array  :  0.48189195379300376
Sorted array   :  None
Sorted array   :  [[0.44297591 0.60293683 0.81751272]
 [0.0450431  0.43862026 0.53680069]
 [0.24909415 0.29692352 0.90712041]]
Sin of array   :  [[0.42863003 0.56706391 0.7294467 ]
 [0.04502787 0.42469073 0.51138929]
 [0.24652617 0.29257974 0.78773313]]
```

```
#Question 1-C
det_arr = np.linalg.det(arr)
print("Determinant of array : ",det_arr)
eig_arr = np.linalg.eig(arr)
print("Eigen values of array : ",eig_arr)
```

```
Determinant of array :  0.08324594150600237
Eigen values of array :  EigResult(eigenvalues=array([1.39984533+0.j
       0.19443562-0.1471827j]), eigenvectors=array([[ 0.72673011+0.j
         0.76644255-0.j        ],
       [ 0.36057213+0.j        , -0.06350413+0.54982181j,
        -0.06350413-0.54982181j],
       [ 0.58468032+0.j        , -0.18617799-0.26751968j,
        -0.18617799+0.26751968j]]))
```

```
#Question 1-D
del(list)
mylist = list(range(1,13))
print(mylist)
print("Type of mylist : ",type(mylist))
myarr = np.array(mylist)
print(myarr)
print("Type of myarr : ",type(myarr))
arr_2d = myarr.reshape(2,-1)
print("\n2D array\n",arr_2d)
arr_3d = myarr.reshape(2,2,-1)
print("\n3D array\n",arr_3d)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]
Type of mylist :  <class 'list'>
[ 1  2  3  4  5  6  7  8  9 10 11 12]
Type of myarr :  <class 'numpy.ndarray'>

2D array
 [[ 1  2  3  4  5  6]
 [ 7  8  9 10 11 12]]

3D array
 [[[ 1  2  3]
  [ 4  5  6]]

 [[ 7  8  9]
  [10 11 12]]]
```

```
#Question 1-E
arr1 = np.zeros((3,3))
print("\nZero Matrix\n",arr1)
arr2 = np.ones((2,3))
print("\nOnes Matrix\n",arr2)
arr3 = np.full((3,3),10)
print("\nSame Matrix\n",arr3)
arr4 = np.eye(3)
print("\nIdentity Matrix\n",arr4)
```

```
arr5 = np.random.randint(0,10,(3,3))
print("\nZero Matrix\n",arr5)
```

```
Zero Matrix
 [[0. 0. 0.]
 [0. 0. 0.]
 [0. 0. 0.]]

Ones Matrix
 [[1. 1. 1.]
 [1. 1. 1.]]

Same Matrix
 [[10 10 10]
 [10 10 10]
 [10 10 10]]

Identity Matrix
 [[1. 0. 0.]
 [0. 1. 0.]
 [0. 0. 1.]]

Zero Matrix
 [[8 7 9]
 [4 0 6]
 [1 0 3]]
```

```
#Question 1-E
det_arr = np.linalg.det(arr)
print("Determinant of array : ",det_arr)
eig_arr = np.linalg.eig(arr)
print("Eigen values of array : ",eig_arr)
```

```
Determinant of array :  0.08324594150600237
Eigen values of array :  EigResult(eigenvalues=array([1.39984533+0.j
       0.19443562-0.1471827j]), eigenvectors=array([[ 0.72673011+0.j
         0.76644255-0.j        ],
       [ 0.36057213+0.j        , -0.06350413+0.54982181j,
        -0.06350413-0.54982181j],
       [ 0.58468032+0.j        , -0.18617799-0.26751968j,
        -0.18617799+0.26751968j]]))
```

```
#Question 2-A
mylist = list(range(10,26))
my_series = pd.Series(mylist)
print(my_series)
print(type(my_series))
```

```
0     10
1     11
2     12
3     13
4     14
5     15
```

```
6     16
7     17
8     18
9     19
10    20
11    21
12    22
13    23
14    24
15    25
dtype: int64
<class 'pandas.core.series.Series'>
```

```
#Question 2-B
print("Index of the series",my_series.index)
print("Value of the series",my_series.values)
```

```
Index of the series RangeIndex(start=0, stop=16, step=1)
Value of the series [10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]
```

```
#Question 2-C
list1 = list(range(1,11))
print(type(list1))
print(mylist)
np_arr = np.array(mylist)
print(type(np_arr))
print(np_arr)
pd_series = pd.Series(np_arr)
print(type(my_series))
print(my_series)
```

```
<class 'list'>
[10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25]
<class 'numpy.ndarray'>
[10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25]
<class 'pandas.core.series.Series'>
0     10
1     11
2     12
3     13
4     14
5     15
6     16
7     17
8     18
9     19
10    20
11    21
12    22
13    23
14    24
15    25
dtype: int64
```

```
#Question 2-D
```

```
list = ['apple','banana','mango']
pd_ser = pd.Series(list,index=[1,2,3])
print(pd_ser)
```

```
1     apple
2    banana
3     mango
dtype: object
```

```
#Question 2-E
print(pd_ser[1])
print(pd_ser[2])
print(pd_ser[3])
```

```
apple
banana
mango
```

```
#Question 2-F
#from google.colab import files
#uploaded = files.upload()

data_read = pd.read_csv("pincode.csv")
print(data_read)
```

```
            pincode        district    statename
0            503110       KAMAREDDY    TELANGANA
1            503145       KAMAREDDY    TELANGANA
2            503122       KAMAREDDY    TELANGANA
3            503122       KAMAREDDY    TELANGANA
4            503112       KAMAREDDY    TELANGANA
...             ...             ...          ...
165627       721153  MEDINIPUR WEST  WEST BENGAL
165628       721629  MEDINIPUR EAST  WEST BENGAL
165629       721631  MEDINIPUR EAST  WEST BENGAL
165630       721152  MEDINIPUR EAST  WEST BENGAL
165631       721154  MEDINIPUR EAST  WEST BENGAL

[165632 rows x 3 columns]
```

```
#Question 3-A
data = {
    'Name': ['Alice', 'Bob', 'Charlie'],
    'Age': [25, 30, 35],
    'Score': [85.5, 90.3, 78.6]
}
df = pd.DataFrame(data)
print(df)
```

```
      Name  Age  Score
0    Alice   25   85.5
1      Bob   30   90.3
2  Charlie   35   78.6
```

```
#Question 3-B
#from google.colab import files
#uploaded = files.upload()

data_read = pd.read_csv("pincode.csv")
print(data_read)
```

```
            pincode        district    statename
    0        503110       KAMAREDDY    TELANGANA
    1        503145       KAMAREDDY    TELANGANA
    2        503122       KAMAREDDY    TELANGANA
    3        503122       KAMAREDDY    TELANGANA
    4        503112       KAMAREDDY    TELANGANA
    ...         ...             ...          ...
    165627   721153  MEDINIPUR WEST  WEST BENGAL
    165628   721629  MEDINIPUR EAST  WEST BENGAL
    165629   721631  MEDINIPUR EAST  WEST BENGAL
    165630   721152  MEDINIPUR EAST  WEST BENGAL
    165631   721154  MEDINIPUR EAST  WEST BENGAL

    [165632 rows x 3 columns]
```

```
#Question 3-C
from sklearn.datasets import load_iris
# Load Iris dataset
iris = load_iris()
# Convert to DataFrame
df_iris = pd.DataFrame(data=iris.data, columns=iris.feature_names)
df_iris['target'] = iris.target
print(df_iris)
```

```
         sepal length (cm)  sepal width (cm)  petal length (cm)  petal width (c
    0                  5.1               3.5                1.4               0
    1                  4.9               3.0                1.4               0
    2                  4.7               3.2                1.3               0
    3                  4.6               3.1                1.5               0
    4                  5.0               3.6                1.4               0
    ..                 ...               ...                ...               .
    145                6.7               3.0                5.2               2
    146                6.3               2.5                5.0               1
    147                6.5               3.0                5.2               2
    148                6.2               3.4                5.4               2
    149                5.9               3.0                5.1               1

         target
    0         0
    1         0
    2         0
    3         0
    4         0
    ..      ...
    145       2
    146       2
    147       2
    148       2
```

```
148       2
149       2

[150 rows x 5 columns]
```