EXPERIMENT NO -3

Roshan Thomas

RMCA-B

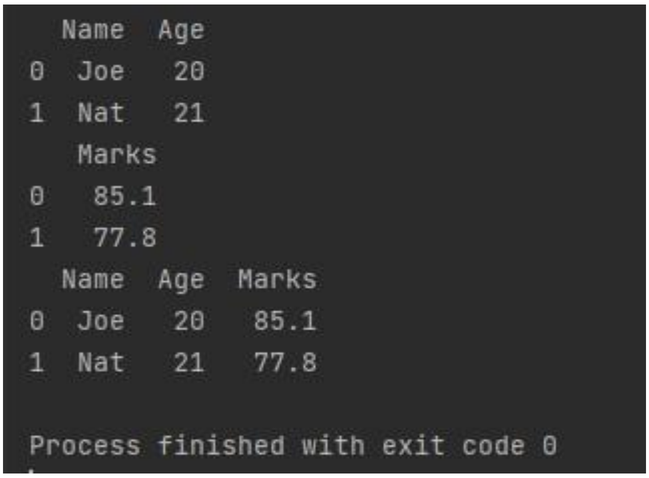
Roll no:15

Date: 25/11/2021

Program-1 import

```
pandas as pd
student_dict ={'Name':['Joe','Nat'],'Age':[20,21]}
student_df      =      pd.DataFrame(student_dict)
print(student_df)
#create dataframe from dict
marks_dict={'Marks':[85.10,77.80]}
marks_df= pd.DataFrame(marks_dict)
print(marks_df) #join Dataframes
joined_df=student_df.join(marks_df) print(joined_df)
```

Output



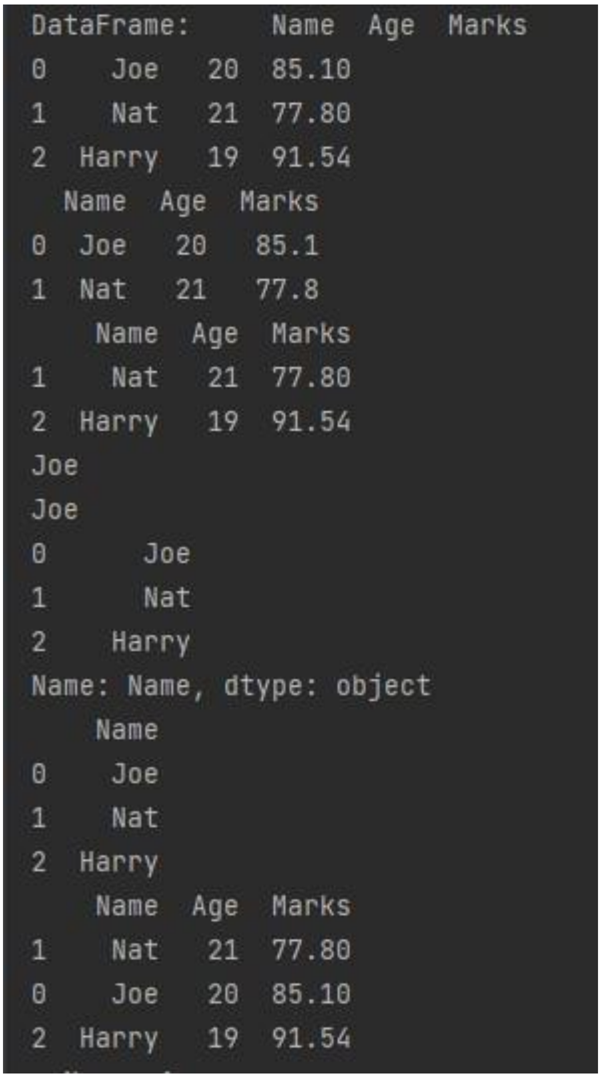
```
  Name  Age
0  Joe   20
1  Nat   21
   Marks
0   85.1
1   77.8
  Name  Age  Marks
0  Joe   20   85.1
1  Nat   21   77.8

Process finished with exit code 0
```

Program-2 import pandas as

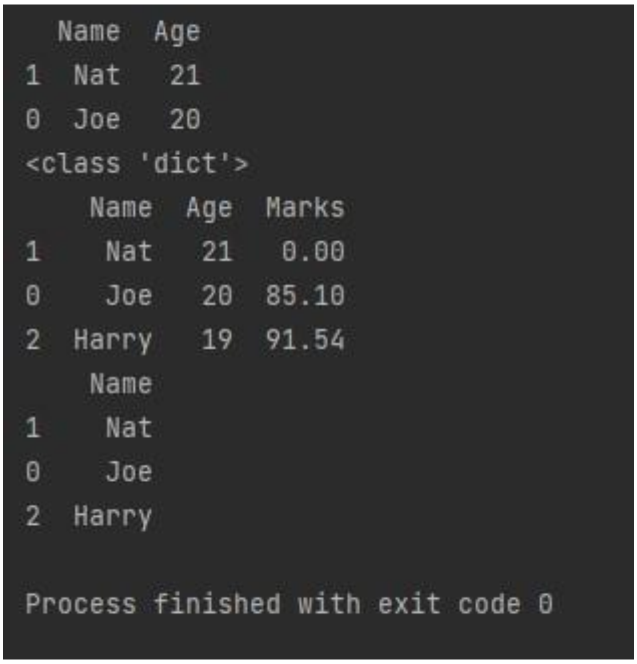
```
pd #Create Dataframe from
dict student_dict
```

```
={'Name':['Joe','Nat','Harry'],'Age':[20,21,19],'Marks':[85.10,77.80,91.54]}
student_df = pd.DataFrame(student_dict)
#Display dataframe
print("DataFrame:",student_df)
#select top 2 rows
print(student_df.head(2)) #select
bottom 2 rows
print(student_df.tail(2))
#select value at row index 0 and column 'Name'
print(student_df.at[0,'Name']) #select value at
first row and column print(student_df.iat[0,0])
#select values of 'Name' column
print(student_df.get('Name'))
#select values from row index 0 to 2 and 'Name' column
print(student_df.loc[0:2,['Name']])
student_df=student_df.sort_values(by=['Marks']) print(student_df)
print(student_df.iloc[0:2,0:2])

print(dict)
filter=student_df['Marks']>80
student_df['Marks'].where(filter,other=0,inplace=True) print(student_df)
student_df=student_df.filter(like='N',axis='columns') print(student_df)
```

Output



```
DataFrame:     Name  Age  Marks
0    Joe   20  85.10
1    Nat   21  77.80
2  Harry   19  91.54
  Name  Age  Marks
0  Joe   20   85.1
1  Nat   21   77.8
    Name  Age  Marks
1    Nat   21  77.80
2  Harry   19  91.54
Joe
Joe
0      Joe
1      Nat
2    Harry
Name: Name, dtype: object
    Name
0    Joe
1    Nat
2  Harry
    Name  Age  Marks
1    Nat   21  77.80
0    Joe   20  85.10
2  Harry   19  91.54
```



```
  Name  Age
1  Nat   21
0  Joe   20
<class 'dict'>
    Name  Age  Marks
1    Nat   21   0.00
0    Joe   20  85.10
2  Harry   19  91.54
    Name
1    Nat
0    Joe
2  Harry

Process finished with exit code 0
```

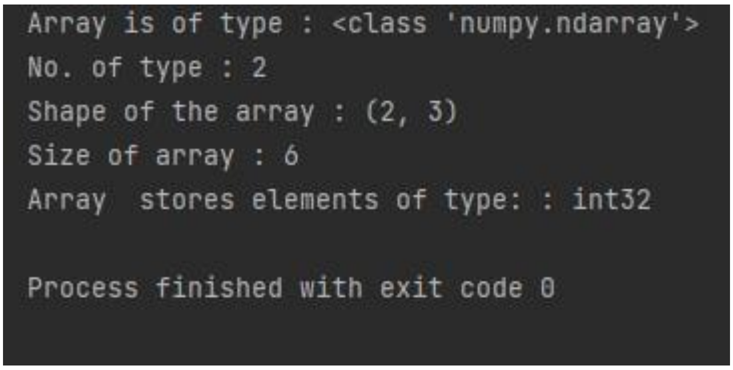
Program-3

Write a python program to demonstrate basic array characterstics

Program import numpy as np #creating array object

```
arr=np.array([[1,2,3],[4,2,5]]) #printing type of arr object
print("Array is of type :",type(arr)) #printing type of arr
dimensions(axes) print("No. of type :",arr.ndim) print("Shape of the
array :",arr.shape)
#printing size of the array print("Size of array
:",arr.size) #printing type of elemmts in array
print("Array  stores elements of type: :",arr.dtype)
```

Output



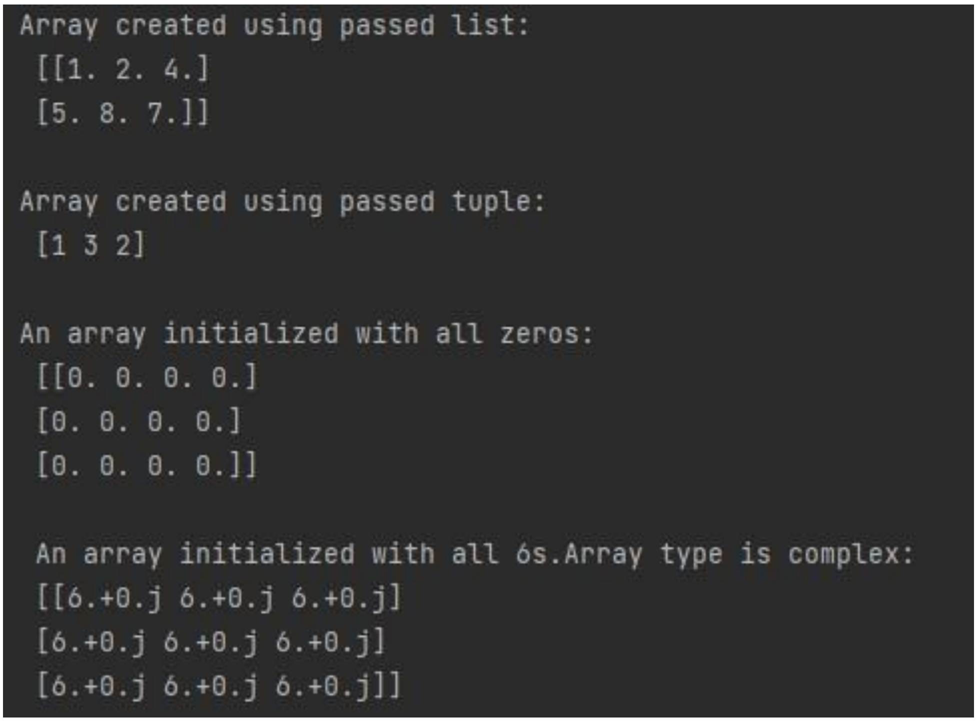
```
Array is of type : <class 'numpy.ndarray'>
No. of type : 2
Shape of the array : (2, 3)
Size of array : 6
Array  stores elements of type: : int32

Process finished with exit code 0
```

Program-4 import

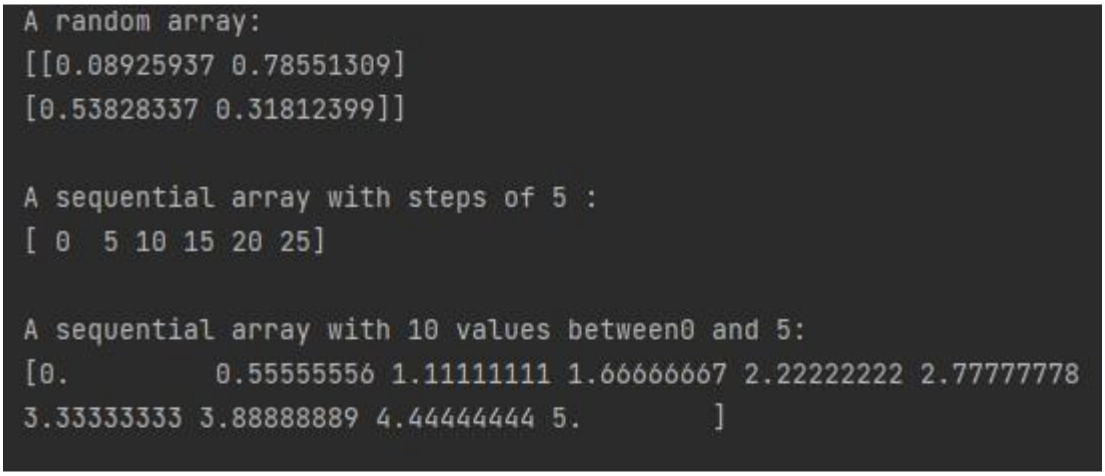
```
numpy as np
#creating array from list with type float
a=np.array([[1,2,4],[5,8,7]],dtype='float') print("Array
created using passed list:\n",a)
#creating array from tuple b=np.array((1,3,2))
print("\nArray created using passed tuple:\n",b)
#creating a 3X4 array with all zeros c=np.zeros((3,4))
print("\nAn array initialized with all zeros:\n",c) #create
a constant value array of complex type
d=np.full((3,3),6,dtype='complex')
print("\n An array initialized with all 6s.""Array type is complex:\n",d)
#create an array with random values
e=np.random.random((2,2))
print("\n A random array:\n",e)
#create a sequence of integers #from
```

```
0 to 30 with steps of 5
f=np.arange(0,30,5)
print("\n A sequential array with steps of 5 :\n",f) #create a
sequence of 10 values in range 0 to 5
g=np.linspace(0,5,10) print("\n A sequential array with 10 values
between""0 and 5:\n",g)
#Reshaping 3x4 array to 2X2X3 array arr=np.array([[1,2,3,4],[5,2,4,2],[1,2,0,1]])
newarr=arr.reshape(2,2,3) print("\nOriginal
array:\n",arr) print("\nReshaped array
:\n",newarr)
#Flattern array
arr=np.array([[1,2,3],[4,5,6]])
flarr=arr.flatten() print("\nOriginal
array:\n",arr) print("\nFlattened
array :\n",flarr)
```

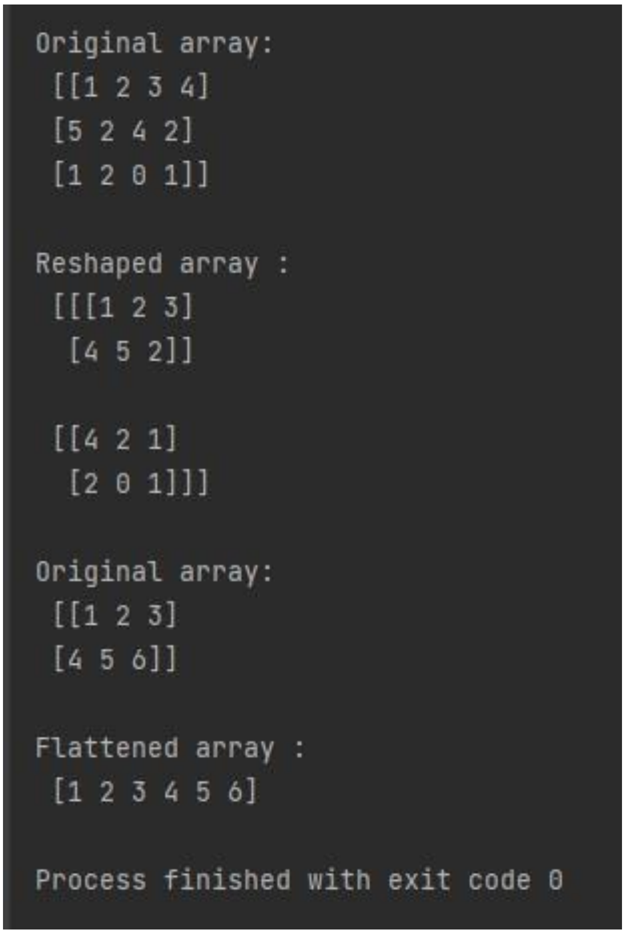
Output



```
Array created using passed list:
 [[1. 2. 4.]
 [5. 8. 7.]]

Array created using passed tuple:
 [1 3 2]

An array initialized with all zeros:
 [[0. 0. 0. 0.]
 [0. 0. 0. 0.]
 [0. 0. 0. 0.]]

 An array initialized with all 6s.Array type is complex:
 [[6.+0.j 6.+0.j 6.+0.j]
 [6.+0.j 6.+0.j 6.+0.j]
 [6.+0.j 6.+0.j 6.+0.j]]
```



```
A random array:
[[0.08925937 0.78551309]
[0.53828337 0.31812399]]

A sequential array with steps of 5 :
[ 0  5 10 15 20 25]

A sequential array with 10 values between0 and 5:
[0.         0.55555556 1.11111111 1.66666667 2.22222222 2.77777778
3.33333333 3.88888889 4.44444444 5.        ]
```



```
Original array:
 [[1 2 3 4]
 [5 2 4 2]
 [1 2 0 1]]

Reshaped array :
 [[[1 2 3]
  [4 5 2]]

 [[4 2 1]
  [2 0 1]]]

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

Flattened array :
 [1 2 3 4 5 6]

Process finished with exit code 0
```

Program-5

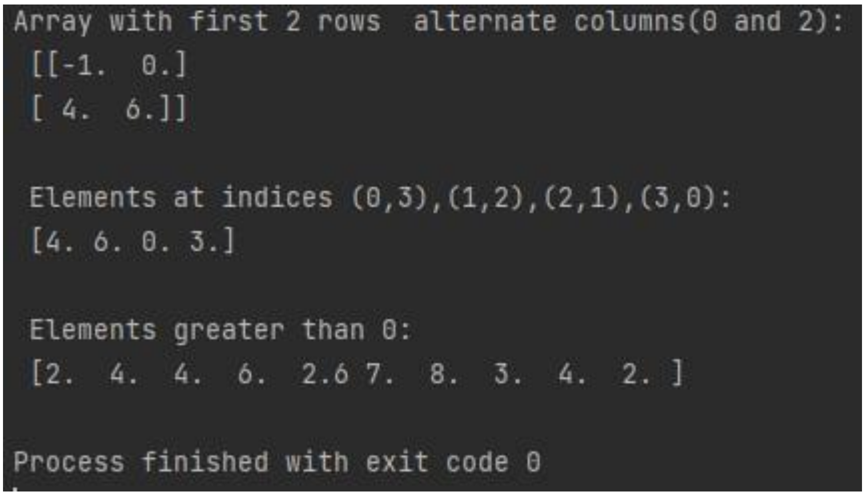
Program to demonstrate indexing in numpy

Program import numpy as np #An exemplar

array

```
arr=np.array([[-1,2,0,4],[4,-0.5,6,0],[2.6,0,7,8],[3,-7,4,2.0]])
#Slicing array temp=arr[:2,::2]
print("Array with first 2 rows  alternate columns(0 and 2):\n",temp)
#Integer array indexing example temp=arr[[0,1,2,3],[3,2,1,0]]
print("\n Elements at indices (0,3),(1,2),(2,1),""(3,0):\n",temp)
#boolean array indexing example cond=arr>0
#cond is a boolean array temp=arr[cond]
print("\n Elements greater than 0:\n",temp)
```

Output



```
Array with first 2 rows  alternate columns(0 and 2):
 [[-1.  0.]
 [ 4.  6.]]

 Elements at indices (0,3),(1,2),(2,1),(3,0):
 [4. 6. 0. 3.]

 Elements greater than 0:
 [2.  4.  4.  6.  2.6 7.  8.  3.  4.  2. ]

Process finished with exit code 0
```

Program to demonstrate basic operations on single array.

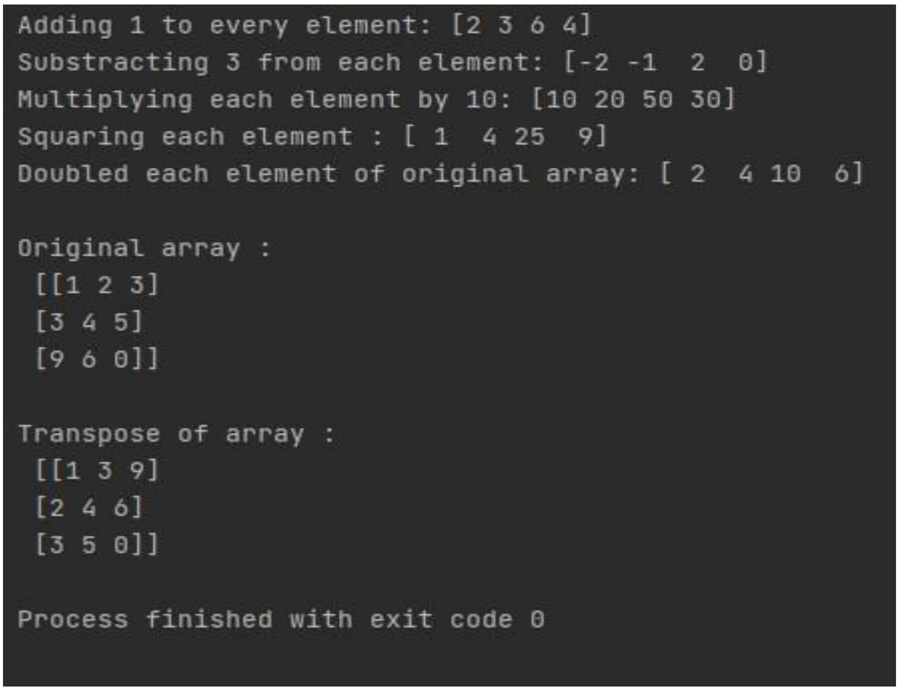
Program

#basic operations on single array.

Program-6

```
import numpy as np
a=np.array([1,2,5,3]) #add
1 to every element
print("Adding 1 to every element:",a+1) #Substracting
3 from each element print("Substracting 3 from each
element:",a-3)
#multipy each element by 10
print("Multiplying each element by 10:",a*10)
#Square each element
print("Squaring each element :",a**2)
#modify existing array
a*=2
print("Doubled each element of original array:",a)
#transpose of array
a=np.array([[1,2,3],[3,4,5],[9,6,0]])
print("\nOriginal array :\n",a) print("\nTranspose
of array :\n",a.T)
```

Output



```
Adding 1 to every element: [2 3 6 4]
Substracting 3 from each element: [-2 -1  2  0]
Multiplying each element by 10: [10 20 50 30]
Squaring each element : [ 1  4 25  9]
Doubled each element of original array: [ 2  4 10  6]

Original array :
 [[1 2 3]
 [3 4 5]
 [9 6 0]]

Transpose of array :
 [[1 3 9]
 [2 4 6]
 [3 5 0]]

Process finished with exit code 0
```

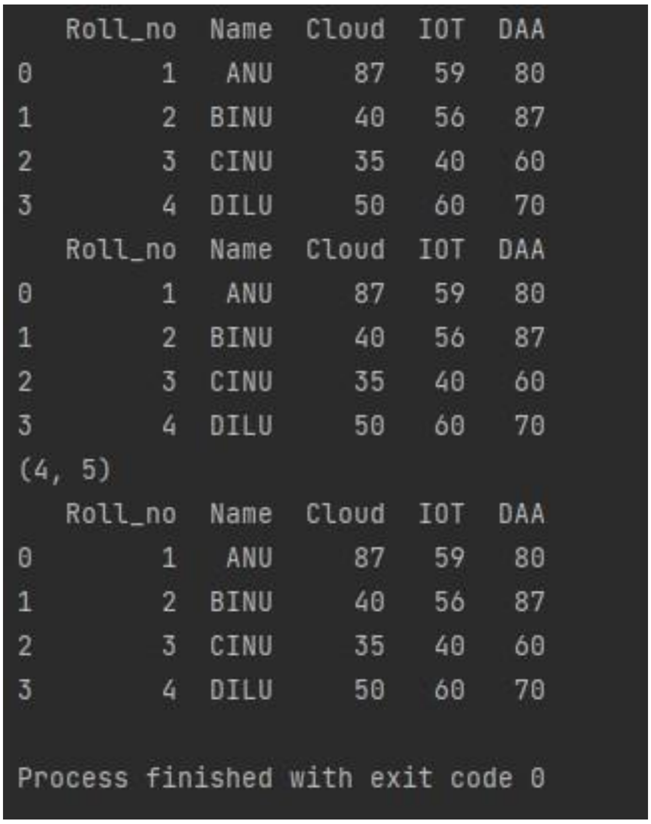
```
        pandas  as pd
df=pd.read_csv("C:/Users/ajcemca/Desktop/Datascience Lab/35_Teena Rose
Mathew/data.csv") print(df.head(5)) print(df.tail(5)) print(df.shape)
print(df.head(5))
```

Program-7

Program

import

Output



```
   Roll_no  Name  Cloud  IOT  DAA
0        1   ANU     87   59   80
1        2  BINU     40   56   87
2        3  CINU     35   40   60
3        4  DILU     50   60   70
   Roll_no  Name  Cloud  IOT  DAA
0        1   ANU     87   59   80
1        2  BINU     40   56   87
2        3  CINU     35   40   60
3        4  DILU     50   60   70
(4, 5)
   Roll_no  Name  Cloud  IOT  DAA
0        1   ANU     87   59   80
1        2  BINU     40   56   87
2        3  CINU     35   40   60
3        4  DILU     50   60   70

Process finished with exit code 0
```

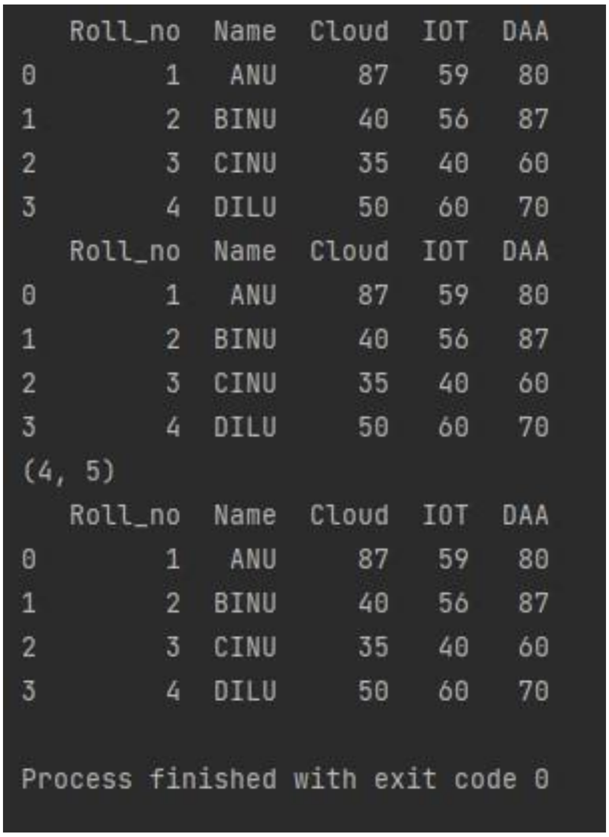
Program-8

Program

import

pandas as pd
df=pd.read_csv("C:/Users/ajcemca/Desktop/Datascience Lab/35_Teena Rose Mathew/data.csv") print(df.head(5)) print(df.tail(5)) print(df.shape)
print(df.head(5))

Output



```
   Roll_no  Name  Cloud  IOT  DAA
0        1   ANU     87   59   80
1        2  BINU     40   56   87
2        3  CINU     35   40   60
3        4  DILU     50   60   70
   Roll_no  Name  Cloud  IOT  DAA
0        1   ANU     87   59   80
1        2  BINU     40   56   87
2        3  CINU     35   40   60
3        4  DILU     50   60   70
(4, 5)
   Roll_no  Name  Cloud  IOT  DAA
0        1   ANU     87   59   80
1        2  BINU     40   56   87
2        3  CINU     35   40   60
3        4  DILU     50   60   70

Process finished with exit code 0
```

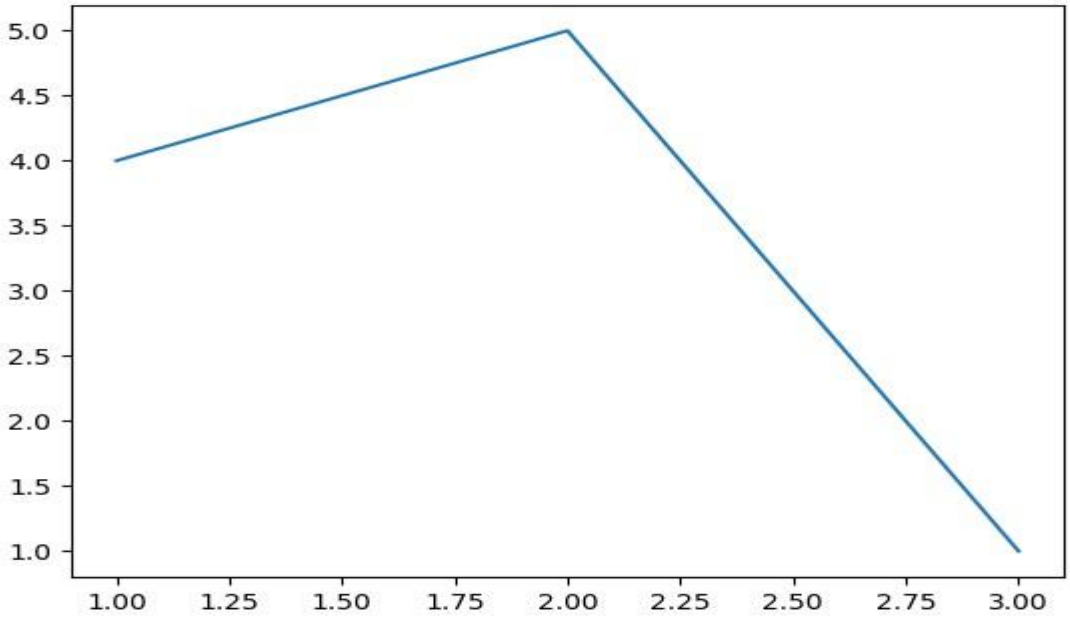
matplotlib.pyplot as plt
import csv

Program-9

Program

```
import



Subjects = []
Scores = []

with open('C:/Users/ajcemca/Desktop/Datascience Lab/35_Teena Rose
Mathew/marks_9.csv', 'r') as csvfile:
    lines = csv.reader(csvfile, delimiter=',')
for row in lines:
        Subjects.append(row[0])
        Scores.append(int(row[1]))

plt.pie(Scores, labels=Subjects, autopct='%.2f%%')
plt.title('Marks of a Student', fontsize=20) plt.show()
```

Output

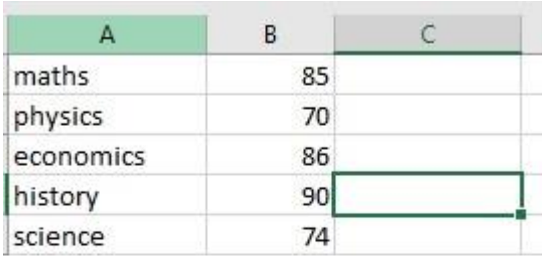


Program-10

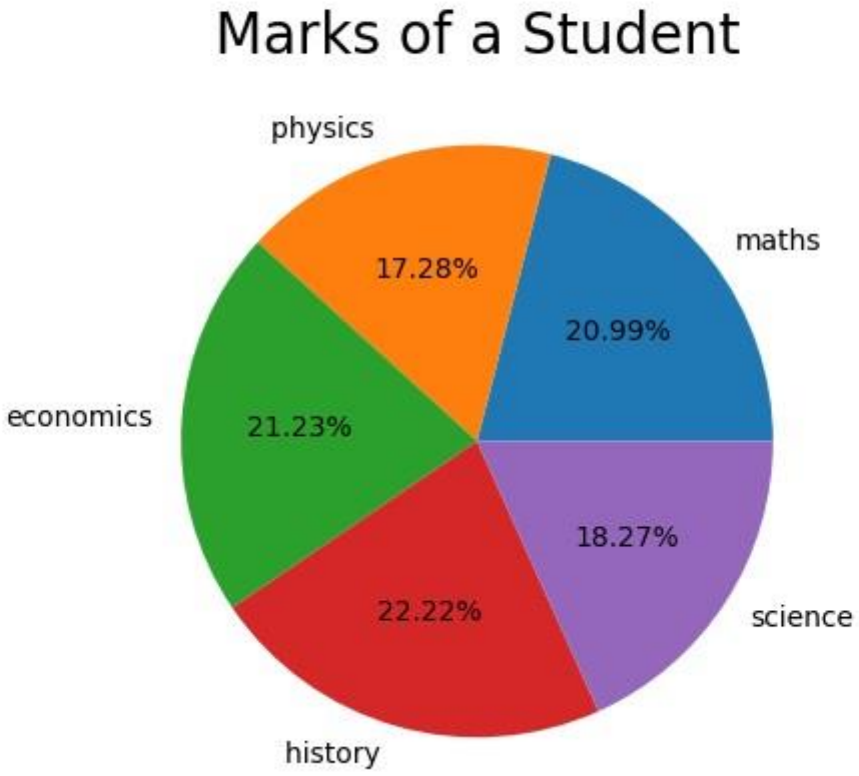
Program

```
from matplotlib import pyplot as plt

# Plotting to our canvas

plt.plot([1, 2, 3], [4, 5, 1])

# Showing what we plotted

plt.show()
```

Output



A	B	C
maths	85	
physics	70	
economics	86	
history	90	
science	74	

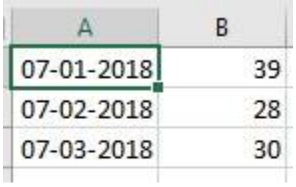


Program-11

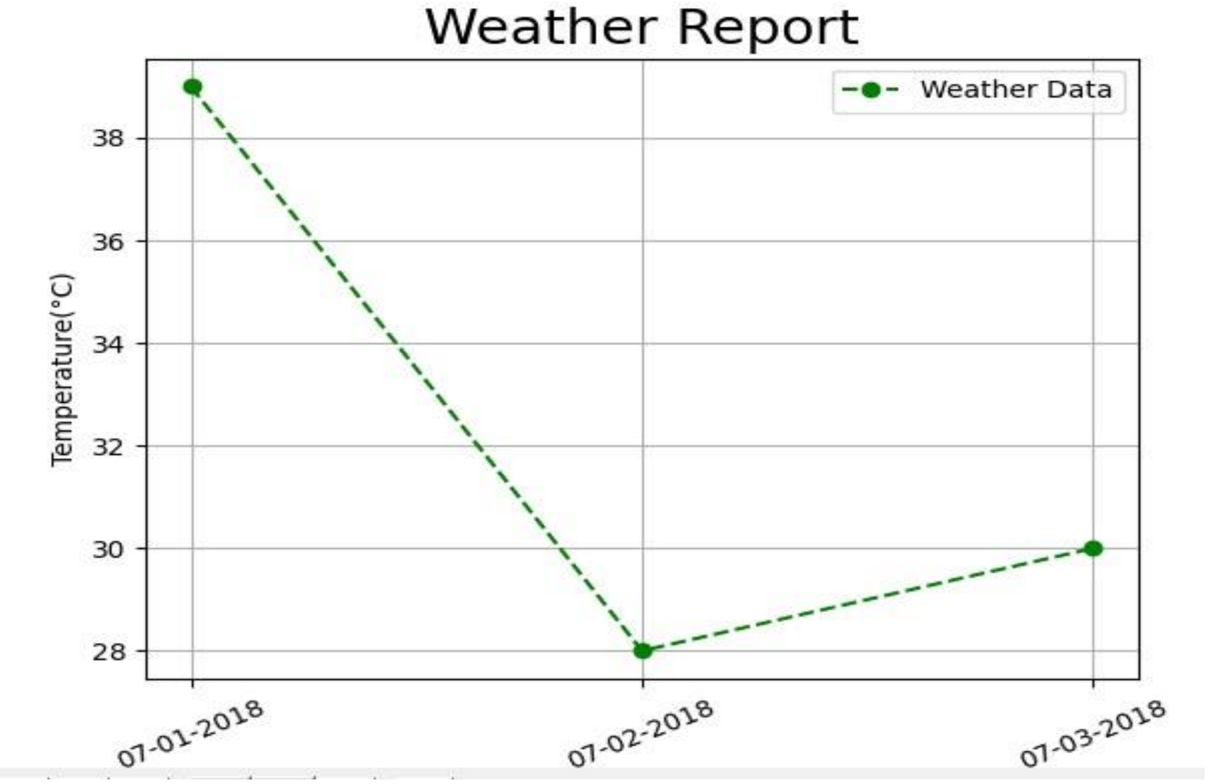
Program

```
import matplotlib.pyplot as plt import
csv

x = [] y
= []

with open('C:/Users/ajcemca/Desktop/Datascience Lab/35_Teena Rose
Mathew/Weather.csv', 'r') as csvfile:     lines = csv.reader(csvfile,
delimiter=',')     for row in lines:
        x.append(row[0])
        y.append(int(row[1]))

plt.plot(x, y, color='g', linestyle='dashed',
         marker='o', label="Weather Data")

plt.xticks(rotation=25) plt.xlabel('Dates')
plt.ylabel('Temperature(°C)')
plt.title('Weather Report', fontsize=20)
plt.grid() plt.legend()
plt.show()
```

Output



A	B
07-01-2018	39
07-02-2018	28
07-03-2018	30


Weather Report
Weather Data
Temperature(°C)
38
36
34
32
30
28
07-01-2018
07-02-2018
07-03-2018