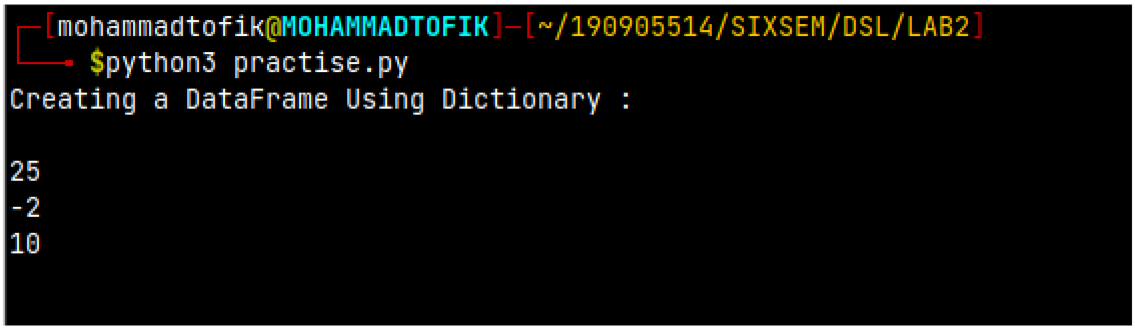
190905514 **MOHAMMAD TOFIK** **BATCH : C2**
ROLLNO : 62 **SECTION : C** **SEM : 6th**

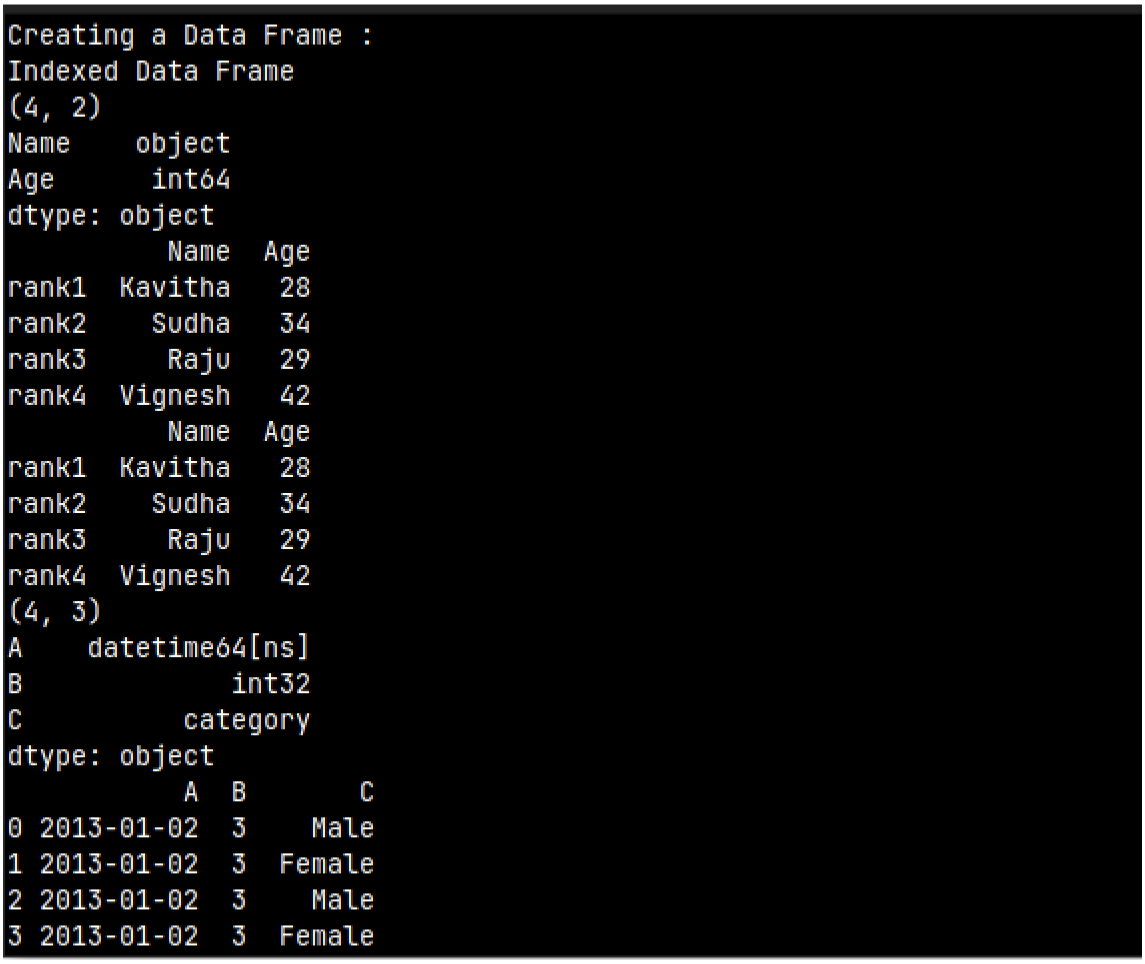
-: DS-LAB-WEEK-2 :-

```
import pandas as pd
import numpy as np

print("Creating a DataFrame Using Dictionary : \n")
s = pd.Series([3, 9, -2, 10, 5])
print(s.sum())
print(s.min())
print(s.max())
```

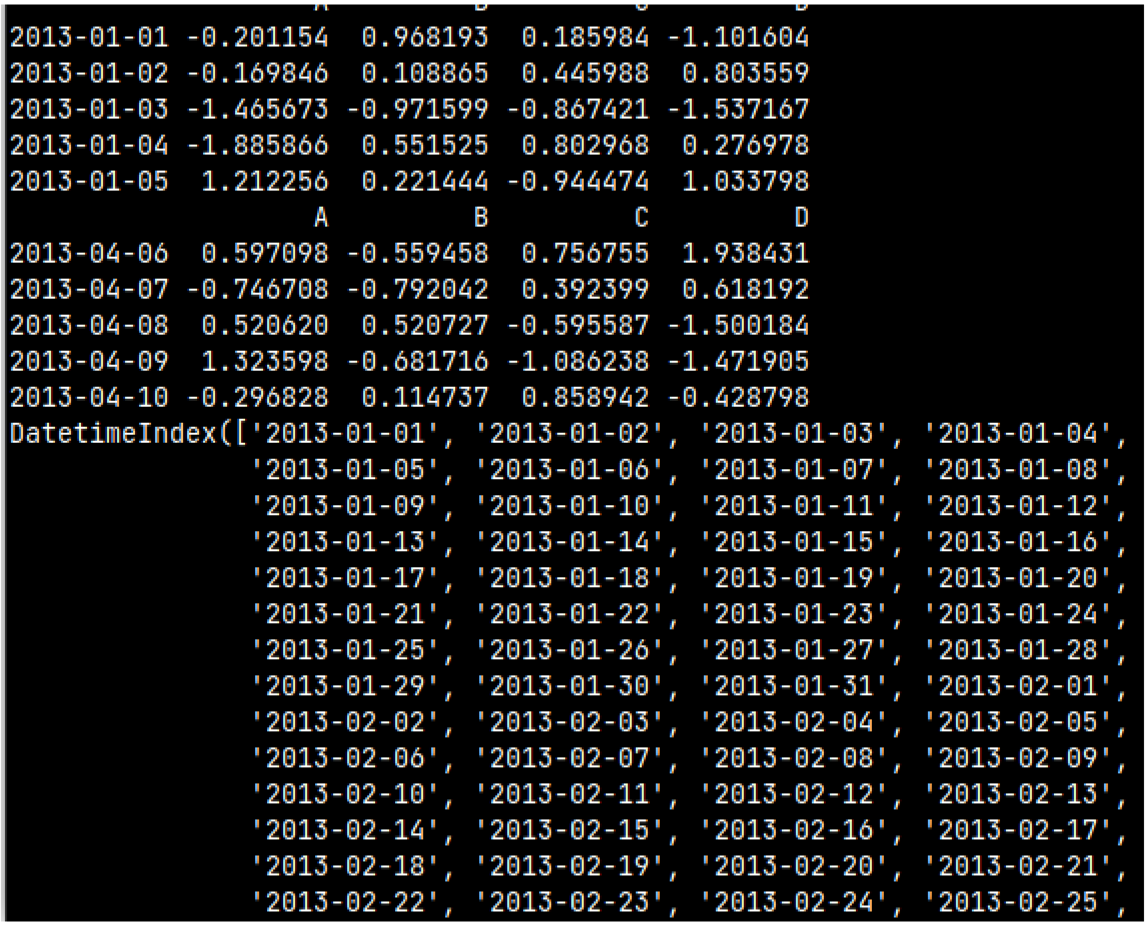


```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 practise.py
Creating a DataFrame Using Dictionary :

25
-2
10
```

```
print("\n")
print("Creating a Data Frame : ")
data = [['Dinesh', 10], ['Nithya', 12], ['Raji', 13]]
df = pd.DataFrame(data, columns=['Name', 'Age'])

print("Indexed Data Frame : ")
data = {'Name': ['Kavitha', 'Sudha', 'Raju',
'Vignesh'], 'Age': [28, 34, 29, 42]}
df = pd.DataFrame(data, index=['rank1', 'rank2', 'rank3', 'rank4'])

print(df.shape)
print(df.dtypes)
print(df.head())
print(df.tail())
df1 = pd.DataFrame({'A': pd.Timestamp('20130102'), 'B': np.array([3]*4,
dtype='int32'),
'C': pd.Categorical(['Male', 'Female', 'Male', 'Female'])})

print(df1.shape)
print(df1.dtypes)
print(df1.head())
print(df1.tail())
```

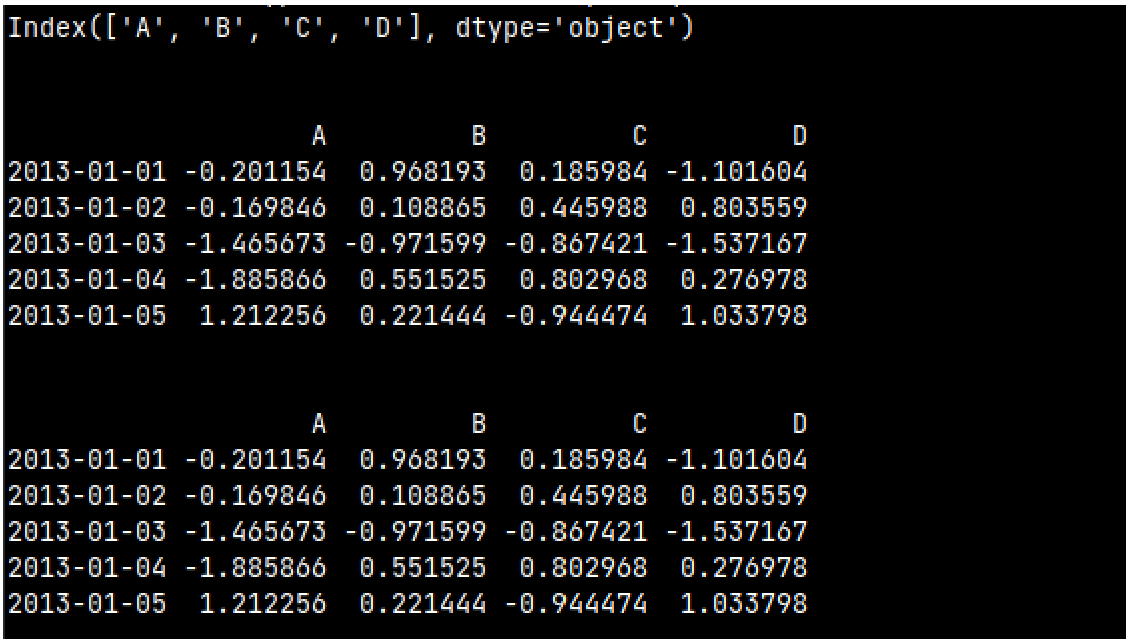


```
Creating a Data Frame :
Indexed Data Frame
(4, 2)
Name    object
Age      int64
dtype: object
          Name  Age
rank1  Kavitha   28
rank2    Sudha   34
rank3     Raju   29
rank4  Vignesh   42
          Name  Age
rank1  Kavitha   28
rank2    Sudha   34
rank3     Raju   29
rank4  Vignesh   42
(4, 3)
A    datetime64[ns]
B             int32
C          category
dtype: object
           A  B       C
0 2013-01-02  3    Male
1 2013-01-02  3  Female
2 2013-01-02  3    Male
3 2013-01-02  3  Female
```



```
2013-01-01 -0.201154  0.968193  0.185984 -1.101604
2013-01-02 -0.169846  0.108865  0.445988  0.803559
2013-01-03 -1.465673 -0.971599 -0.867421 -1.537167
2013-01-04 -1.885866  0.551525  0.802968  0.276978
2013-01-05  1.212256  0.221444 -0.944474  1.033798
                   A         B         C         D
2013-04-06  0.597098 -0.559458  0.756755  1.938431
2013-04-07 -0.746708 -0.792042  0.392399  0.618192
2013-04-08  0.520620  0.520727 -0.595587 -1.500184
2013-04-09  1.323598 -0.681716 -1.086238 -1.471905
2013-04-10 -0.296828  0.114737  0.858942 -0.428798
DatetimeIndex(['2013-01-01', '2013-01-02', '2013-01-03', '2013-01-04',
               '2013-01-05', '2013-01-06', '2013-01-07', '2013-01-08',
               '2013-01-09', '2013-01-10', '2013-01-11', '2013-01-12',
               '2013-01-13', '2013-01-14', '2013-01-15', '2013-01-16',
               '2013-01-17', '2013-01-18', '2013-01-19', '2013-01-20',
               '2013-01-21', '2013-01-22', '2013-01-23', '2013-01-24',
               '2013-01-25', '2013-01-26', '2013-01-27', '2013-01-28',
               '2013-01-29', '2013-01-30', '2013-01-31', '2013-02-01',
               '2013-02-02', '2013-02-03', '2013-02-04', '2013-02-05',
               '2013-02-06', '2013-02-07', '2013-02-08', '2013-02-09',
               '2013-02-10', '2013-02-11', '2013-02-12', '2013-02-13',
               '2013-02-14', '2013-02-15', '2013-02-16', '2013-02-17',
               '2013-02-18', '2013-02-19', '2013-02-20', '2013-02-21',
               '2013-02-22', '2013-02-23', '2013-02-24', '2013-02-25',
```

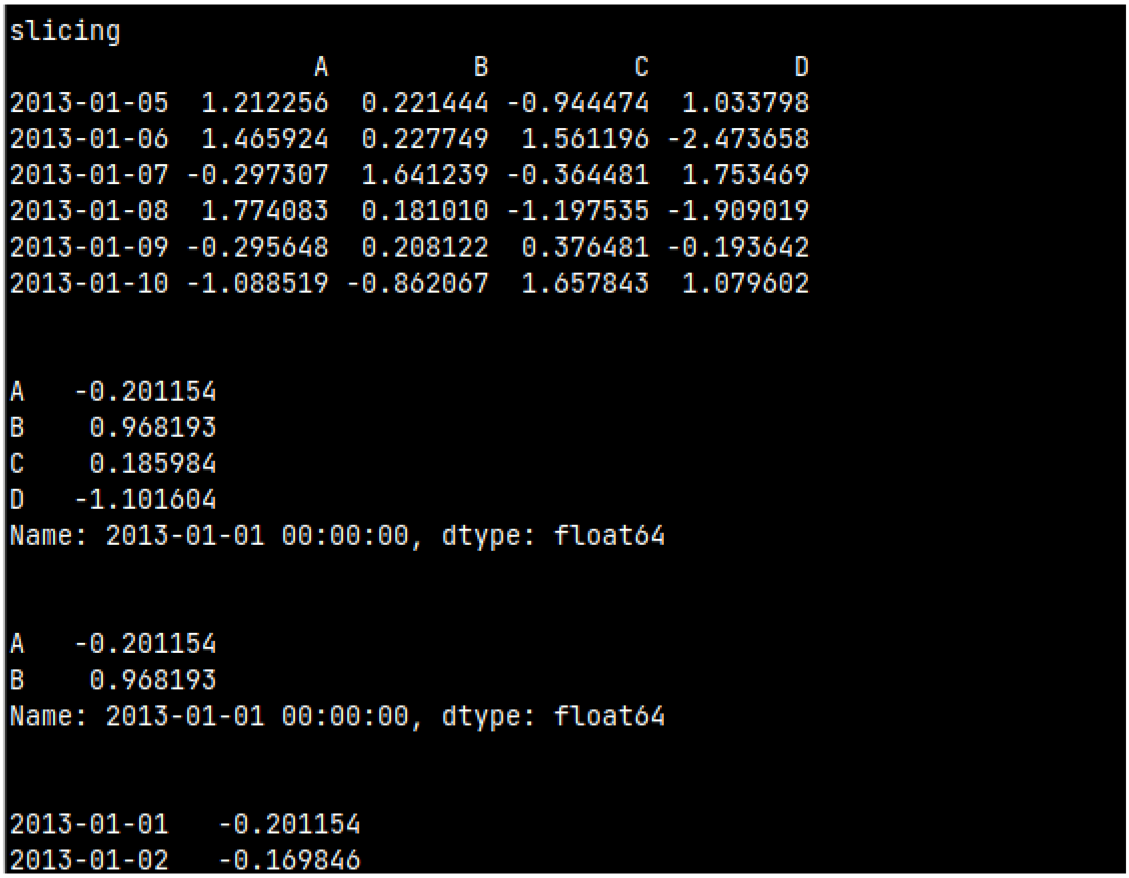
```python
print("\n")
dates = pd.date_range('20130101', periods=100)
df = pd.DataFrame(np.random.randn(100, 4), index=dates, columns=list('ABCD'))

print(df.head())
print(df.tail())
print(df.index)
print(df.columns)
print("\n")
df.sort_index(axis=1, ascending=False)
print(df.head())
print("\n")
df.sort_values(by='B')
print(df.head())
```



```
Index(['A', 'B', 'C', 'D'], dtype='object')


                   A         B         C         D
2013-01-01 -0.201154  0.968193  0.185984 -1.101604
2013-01-02 -0.169846  0.108865  0.445988  0.803559
2013-01-03 -1.465673 -0.971599 -0.867421 -1.537167
2013-01-04 -1.885866  0.551525  0.802968  0.276978
2013-01-05  1.212256  0.221444 -0.944474  1.033798


                   A         B         C         D
2013-01-01 -0.201154  0.968193  0.185984 -1.101604
2013-01-02 -0.169846  0.108865  0.445988  0.803559
2013-01-03 -1.465673 -0.971599 -0.867421 -1.537167
2013-01-04 -1.885866  0.551525  0.802968  0.276978
2013-01-05  1.212256  0.221444 -0.944474  1.033798
```

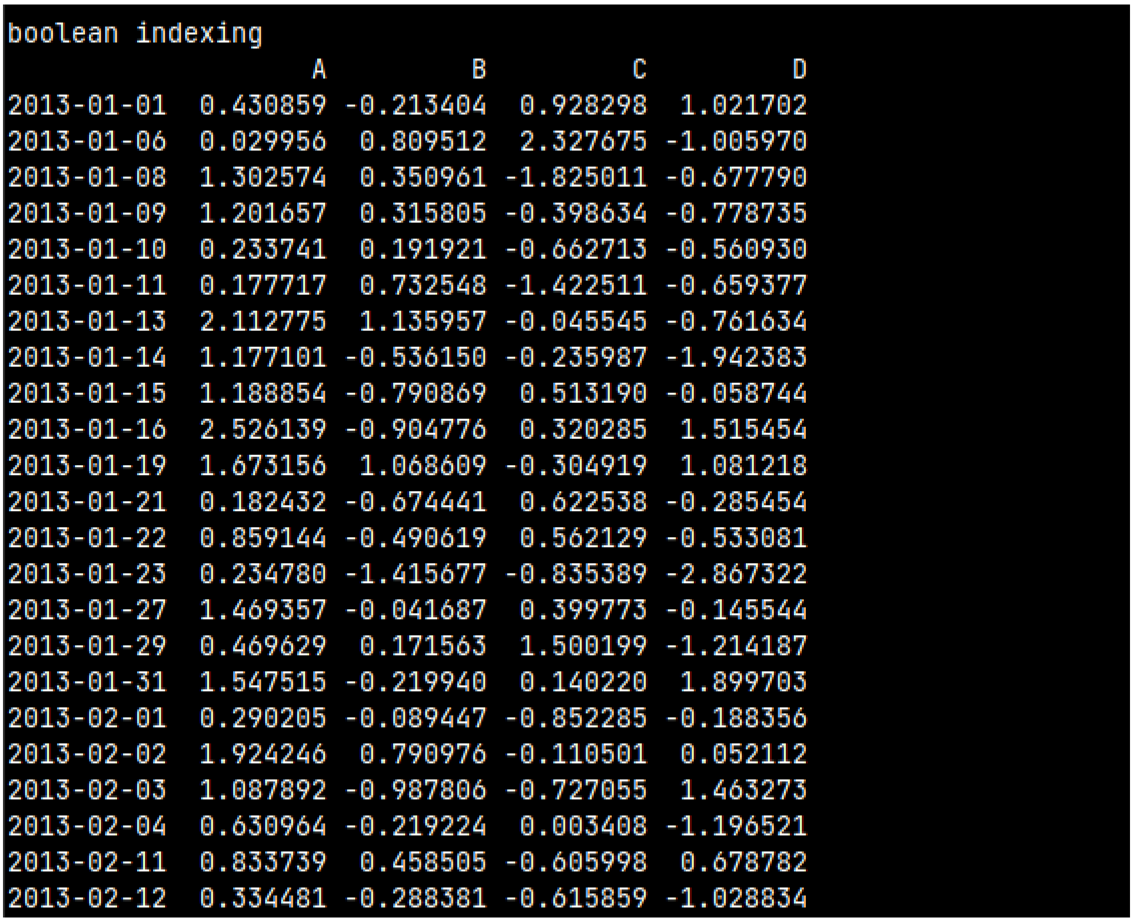
```python
print("\nslicing")
print(df['20130105':'20130110'])

print("\n")
print(df.iloc[0])
print("\n")
print(df.iloc[0, :2])
print("\n")
print(df['A'])
dates = pd.date_range('20130101', periods=100)
df = pd.DataFrame(np.random.randn(100, 4), index=dates, columns=list('ABCD'))
df1 = pd.DataFrame({'A': pd.Timestamp('20130102'), 'B': np.array([3]*4,
dtype='int32'),
'C': pd.Categorical(['Male', 'Female', 'Male', 'Female'])})
```

```
df2 = pd.DataFrame({'A': pd.Timestamp('20130102'), 'B': np.array([3]*4, 
dtype='int32'),
'C': pd.Categorical(['Male', 'Female', 'Male', 'Female'])})
```



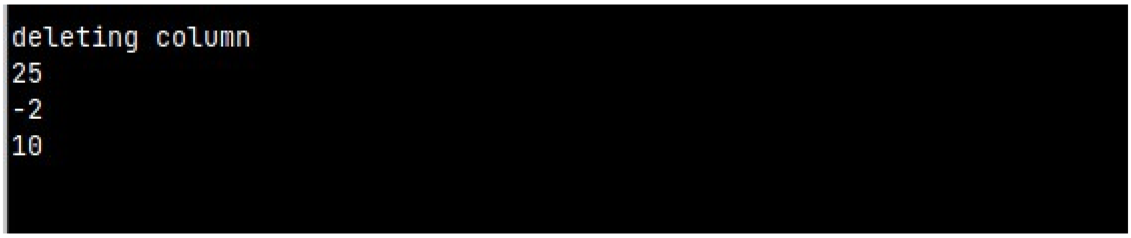
```
slicing
                   A         B         C         D
2013-01-05  1.212256  0.221444 -0.944474  1.033798
2013-01-06  1.465924  0.227749  1.561196 -2.473658
2013-01-07 -0.297307  1.641239 -0.364481  1.753469
2013-01-08  1.774083  0.181010 -1.197535 -1.909019
2013-01-09 -0.295648  0.208122  0.376481 -0.193642
2013-01-10 -1.088519 -0.862067  1.657843  1.079602


A   -0.201154
B    0.968193
C    0.185984
D   -1.101604
Name: 2013-01-01 00:00:00, dtype: float64


A   -0.201154
B    0.968193
Name: 2013-01-01 00:00:00, dtype: float64


2013-01-01   -0.201154
2013-01-02   -0.169846
```

```
print("\nboolean indexing : ")
print(df[df.A > 0])

print("\n")
df.loc[:, 'D'] = np.array([5]*len(df))
```



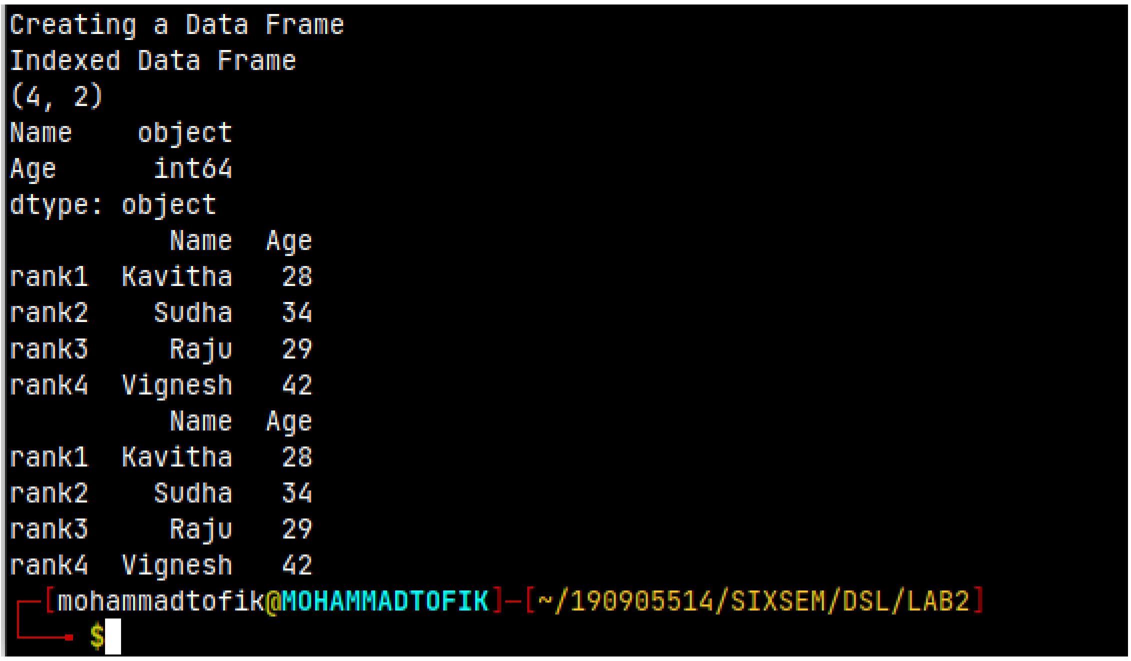
```
boolean indexing
                   A         B         C         D
2013-01-01  0.430859 -0.213404  0.928298  1.021702
2013-01-06  0.029956  0.809512  2.327675 -1.005970
2013-01-08  1.302574  0.350961 -1.825011 -0.677790
2013-01-09  1.201657  0.315805 -0.398634 -0.778735
2013-01-10  0.233741  0.191921 -0.662713 -0.560930
2013-01-11  0.177717  0.732548 -1.422511 -0.659377
2013-01-13  2.112775  1.135957 -0.045545 -0.761634
2013-01-14  1.177101 -0.536150 -0.235987 -1.942383
2013-01-15  1.188854 -0.790869  0.513190 -0.058744
2013-01-16  2.526139 -0.904776  0.320285  1.515454
2013-01-19  1.673156  1.068609 -0.304919  1.081218
2013-01-21  0.182432 -0.674441  0.622538 -0.285454
2013-01-22  0.859144 -0.490619  0.562129 -0.533081
2013-01-23  0.234780 -1.415677 -0.835389 -2.867322
2013-01-27  1.469357 -0.041687  0.399773 -0.145544
2013-01-29  0.469629  0.171563  1.500199 -1.214187
2013-01-31  1.547515 -0.219940  0.140220  1.899703
2013-02-01  0.290205 -0.089447 -0.852285 -0.188356
2013-02-02  1.924246  0.790976 -0.110501  0.052112
2013-02-03  1.087892 -0.987806 -0.727055  1.463273
2013-02-04  0.630964 -0.219224  0.003408 -1.196521
2013-02-11  0.833739  0.458505 -0.605998  0.678782
2013-02-12  0.334481 -0.288381 -0.615859 -1.028834
```

```
print("\ndeleting column : ")
df.drop('A', axis=1, inplace=True)

s=pd.Series([3,9,-2,10,5])
print(s.sum())
print(s.min())
print(s.max())
```



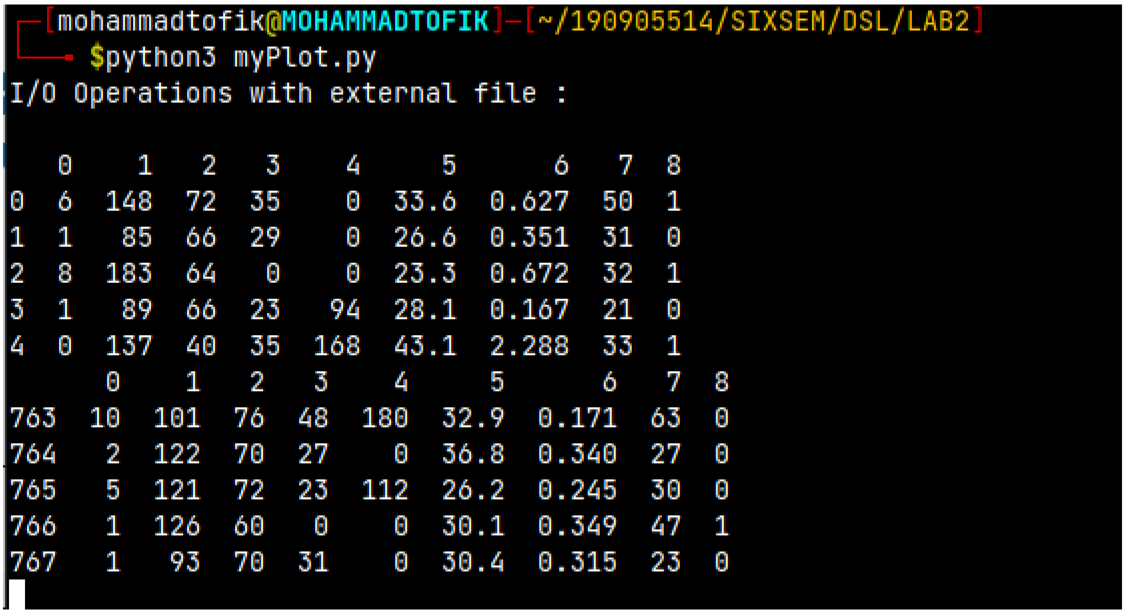
```
deleting column
25
-2
10
```

```
print("\n")
print("Creating a Data Frame : ")
data = [['Dinesh',10],['Nithya',12],['Raji',13]]
df = pd.DataFrame(data,columns=['Name','Age'])

print("Indexed Data Frame")
data = {'Name':['Kavitha', 'Sudha', 'Raju','Vignesh'],'Age':[28,34,29,42]}
df = pd.DataFrame(data, index=['rank1','rank2','rank3','rank4'])

print(df.shape)
print(df.dtypes)
print(df.head())
print(df.tail())
```

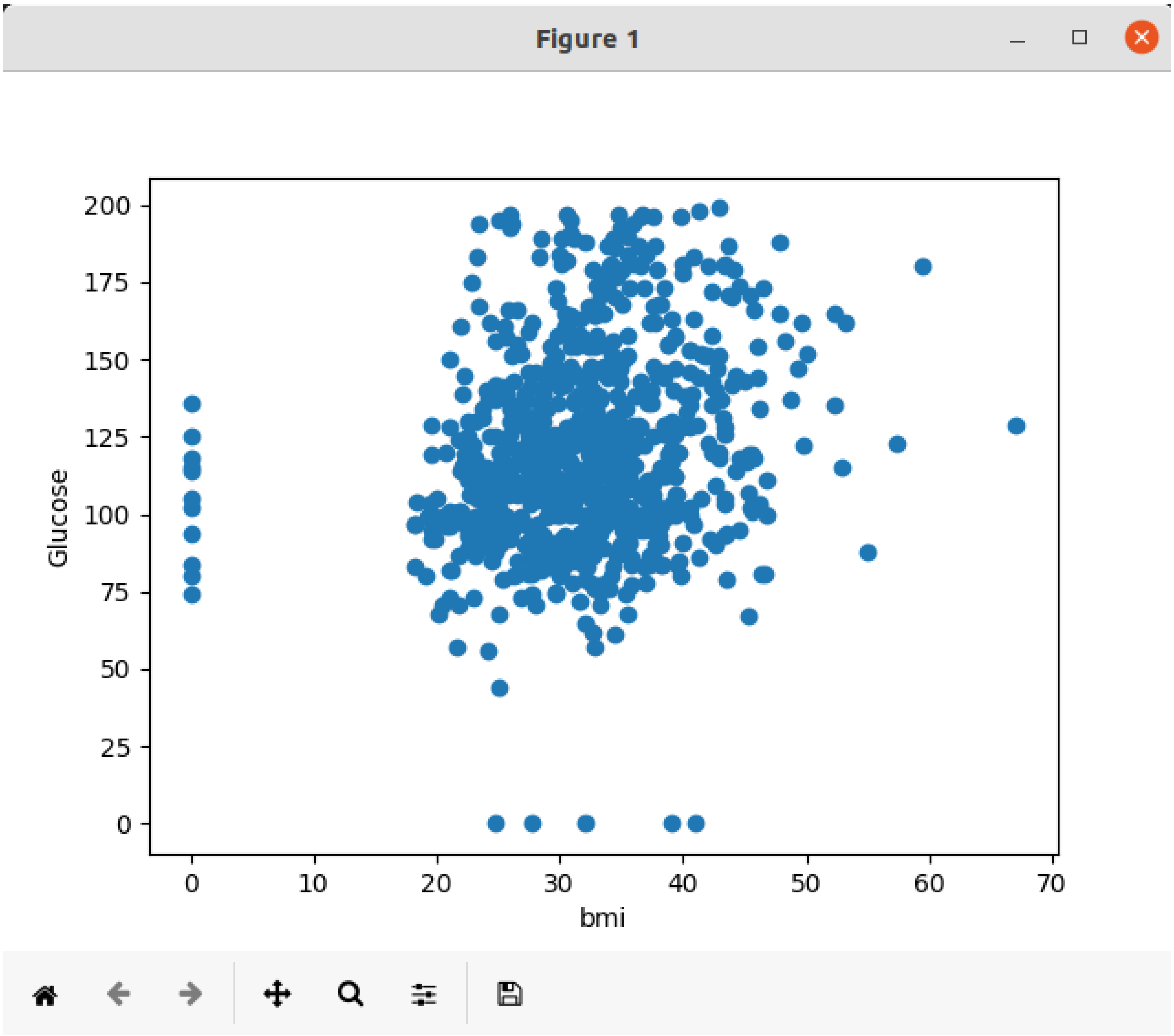


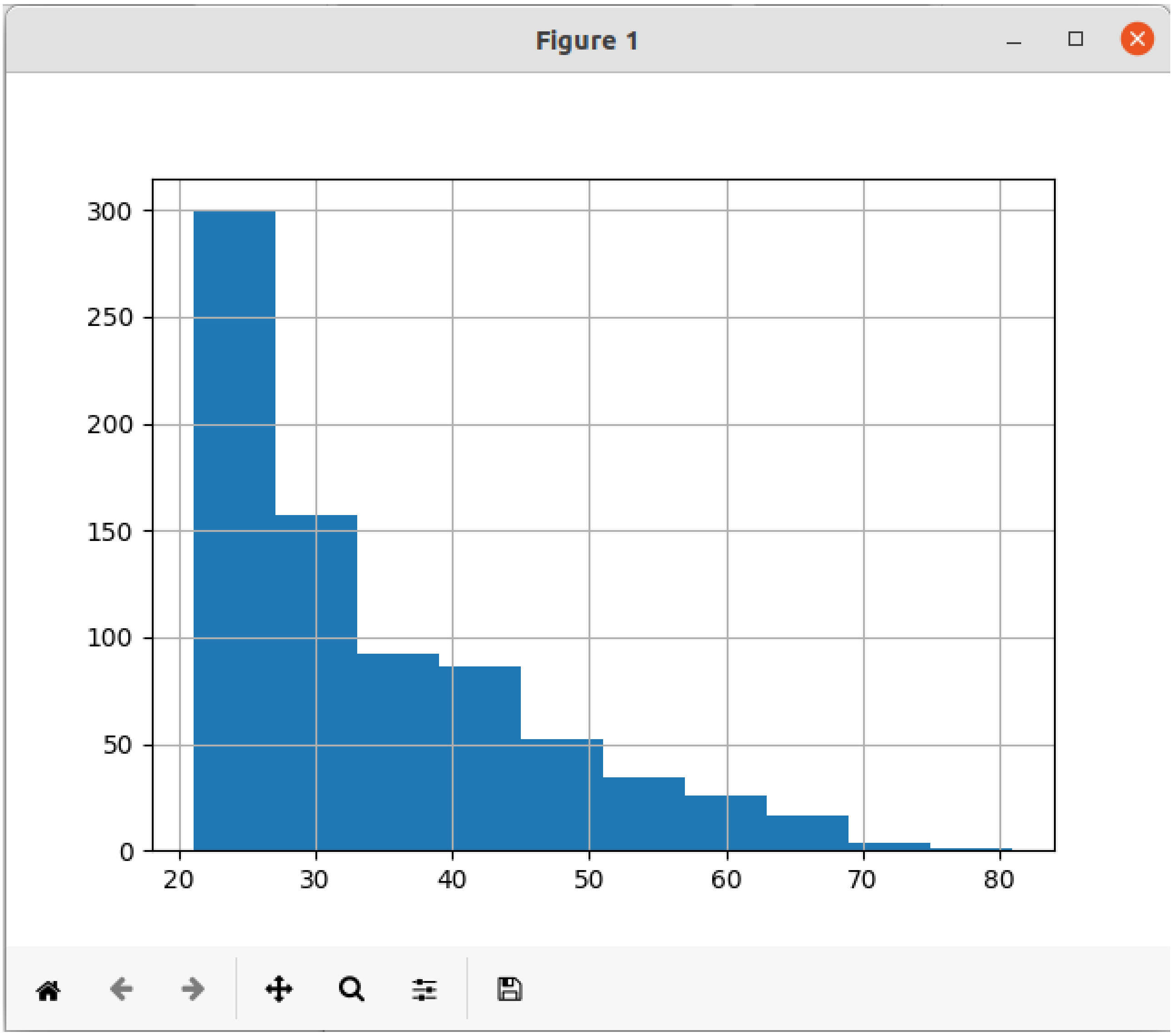
```
Creating a Data Frame
Indexed Data Frame
(4, 2)
Name     object
Age       int64
dtype: object
          Name  Age
rank1  Kavitha   28
rank2    Sudha   34
rank3     Raju   29
rank4  Vignesh   42
          Name  Age
rank1  Kavitha   28
rank2    Sudha   34
rank3     Raju   29
rank4  Vignesh   42
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $
```

```
import pandas as pd
import numpy as numpy
import matplotlib.pyplot as plot
print("I/O Operations with external file : \n")
df = pd.read_csv('xyz.csv', header = None)
print(df.head())
print(df.tail())
df.columns = ['preg', 'glu', 'bp', 'sft', 'ins', 'bmi', 'dpf', 'age', 'class']
plot.scatter(df['bmi'], df['glu'])
plot.xlabel('bmi')
plot.ylabel('Glucose')
plot.show()
df['age'].hist()
plot.show()
f.plot(kind ='bar')
plot.show()
```

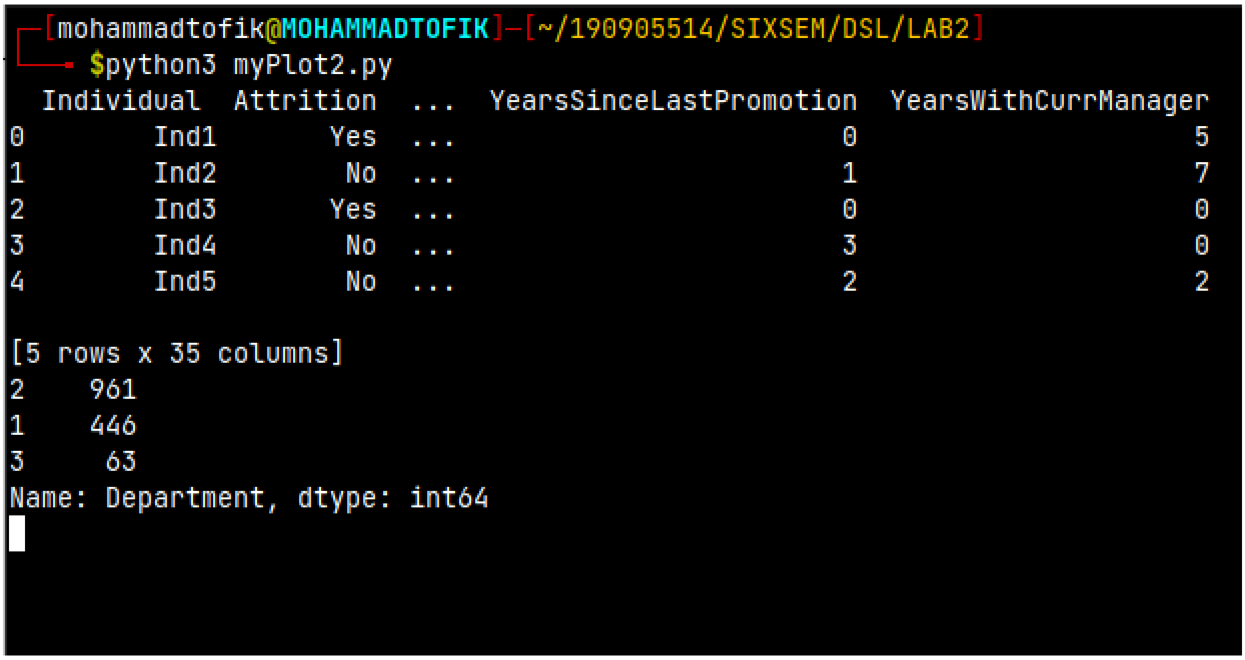


```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 myPlot.py
I/O Operations with external file :

   0    1   2   3    4     5      6   7  8
0  6  148  72  35    0  33.6  0.627  50  1
1  1   85  66  29    0  26.6  0.351  31  0
2  8  183  64   0    0  23.3  0.672  32  1
3  1   89  66  23   94  28.1  0.167  21  0
4  0  137  40  35  168  43.1  2.288  33  1
      0    1   2   3    4     5      6   7  8
763  10  101  76  48  180  32.9  0.171  63  0
764   2  122  70  27    0  36.8  0.340  27  0
765   5  121  72  23  112  26.2  0.245  30  0
766   1  126  60   0    0  30.1  0.349  47  1
767   1   93  70  31    0  30.4  0.315  23  0
```

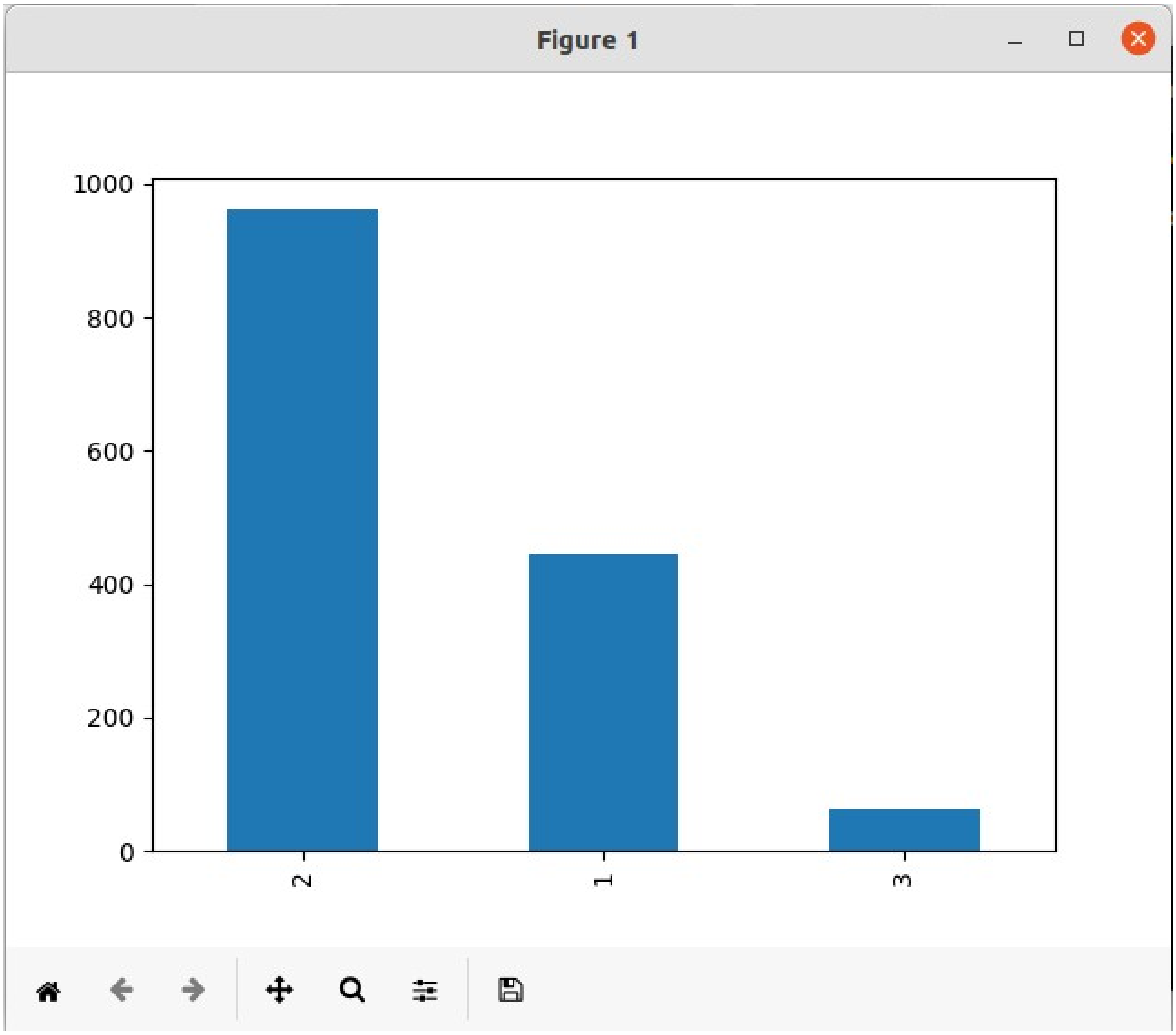




```
import pandas as pd
import numpy as numpy
import matplotlib.pyplot as plot

df = pd.read_table("myTxtFile.txt")
print(df.head())
f = df["Department"].value_counts()
print(f)
f.plot(kind = 'bar')
plot.show()
f.plot(kind = 'pie')
plot.show()
foo = pd.crosstab(df['Gender'], df['Attrition'])
foo.plot(kind ='bar')
plot.show();
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 myPlot2.py
  Individual Attrition  ...  YearsSinceLastPromotion  YearsWithCurrManager
0       Ind1       Yes  ...                        0                     5
1       Ind2        No  ...                        1                     7
2       Ind3       Yes  ...                        0                     0
3       Ind4        No  ...                        3                     0
4       Ind5        No  ...                        2                     2

[5 rows x 35 columns]
2    961
1    446
3     63
Name: Department, dtype: int64
```



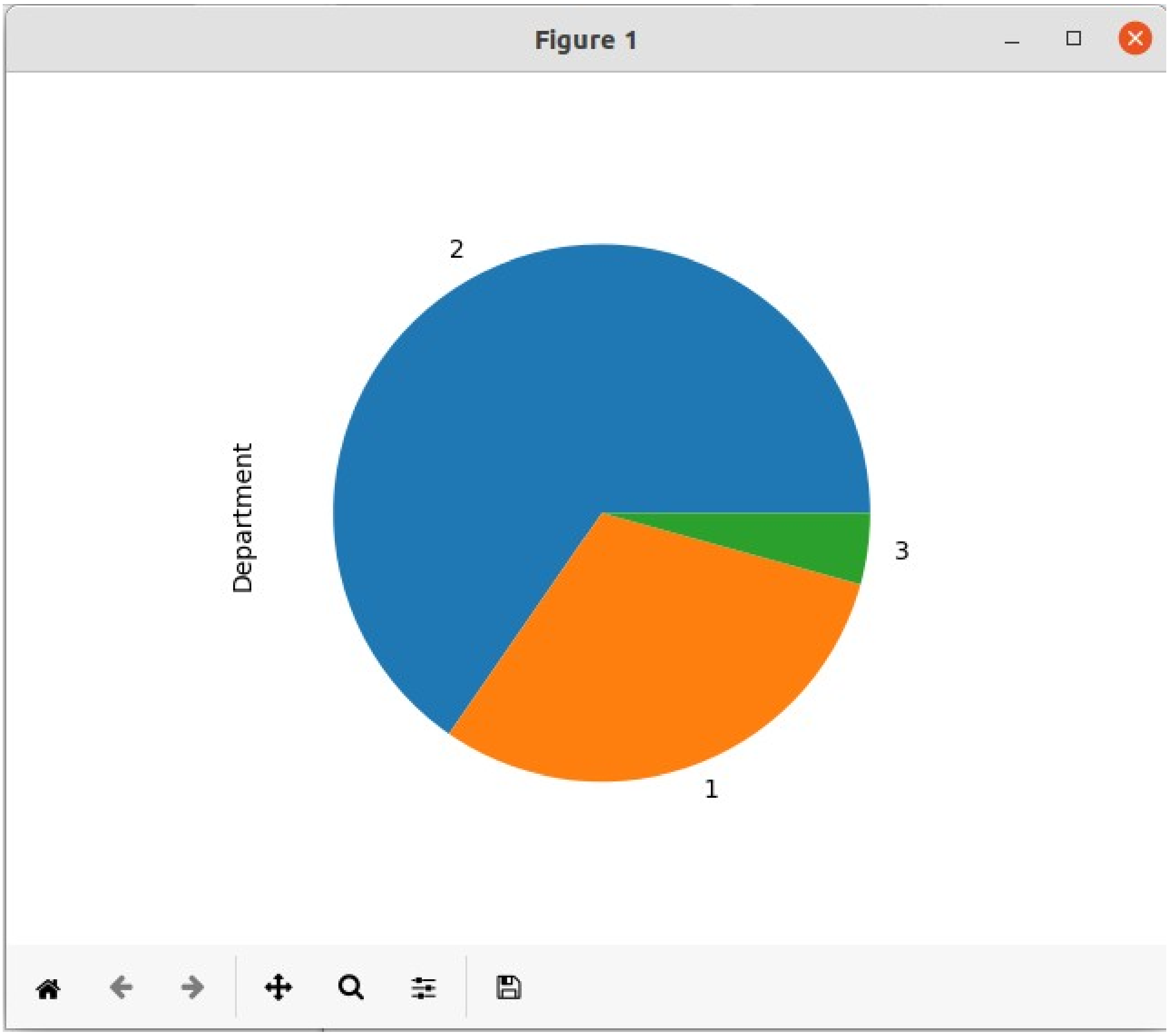


Figure 1
2
Department
3
1

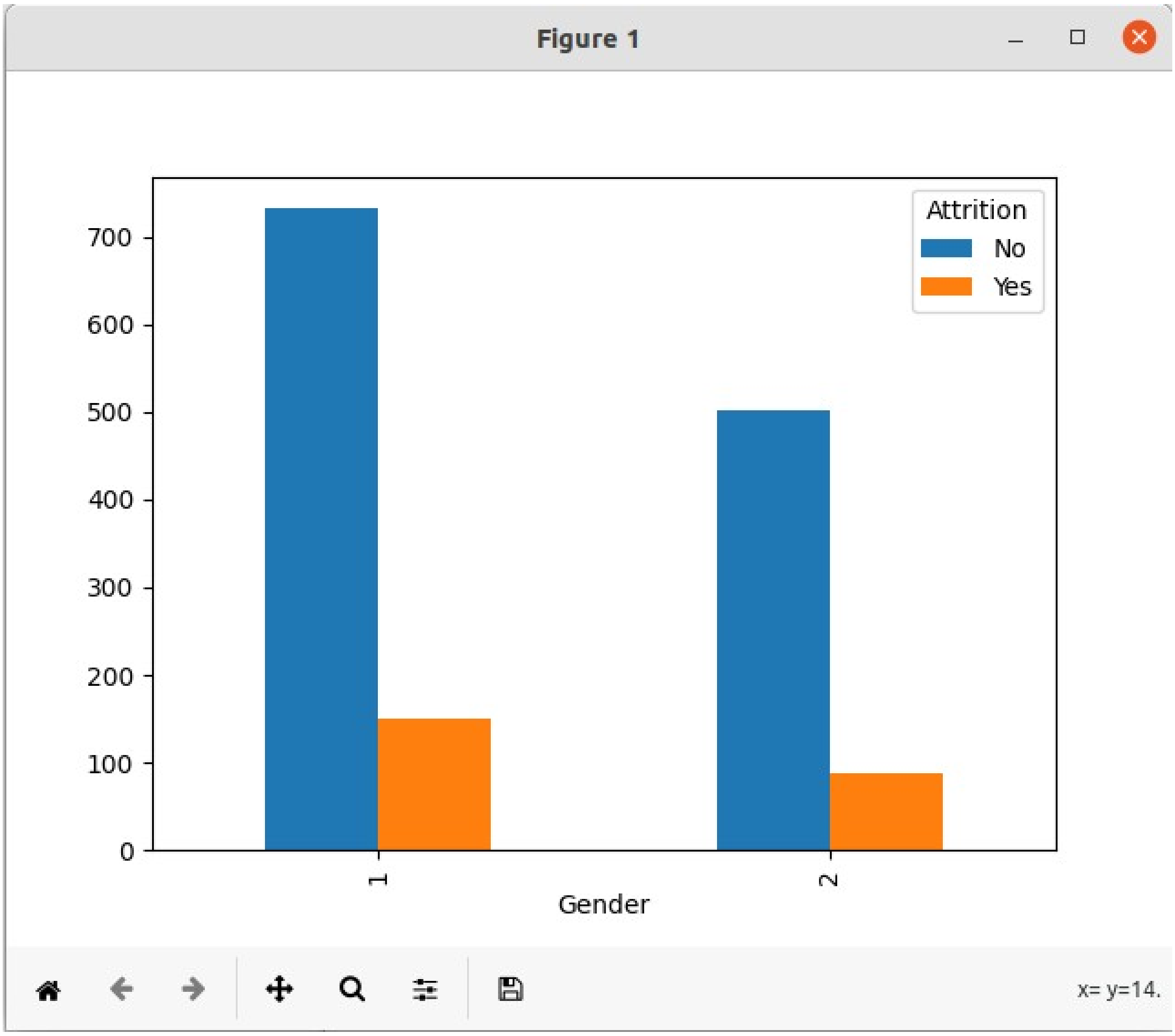
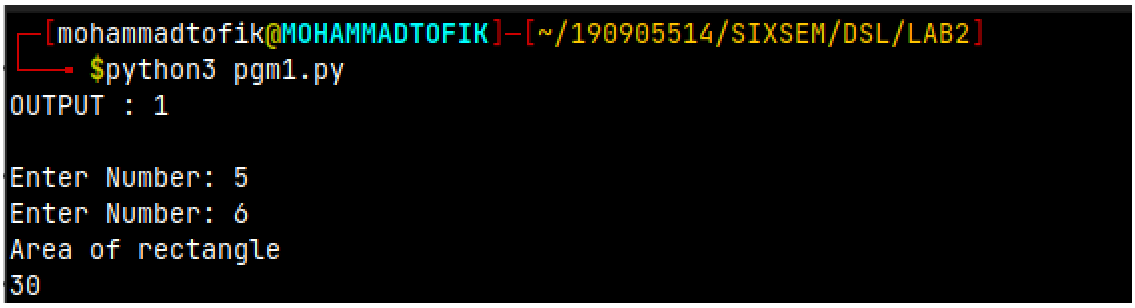


Figure 1
Attrition
No
Yes
700
600
500
400
300
200
100
0
1
2
Gender
x= y=14.

EXCERCISE QUESTIONS :

1.write a program to find the area of reatangle.Take input from the user

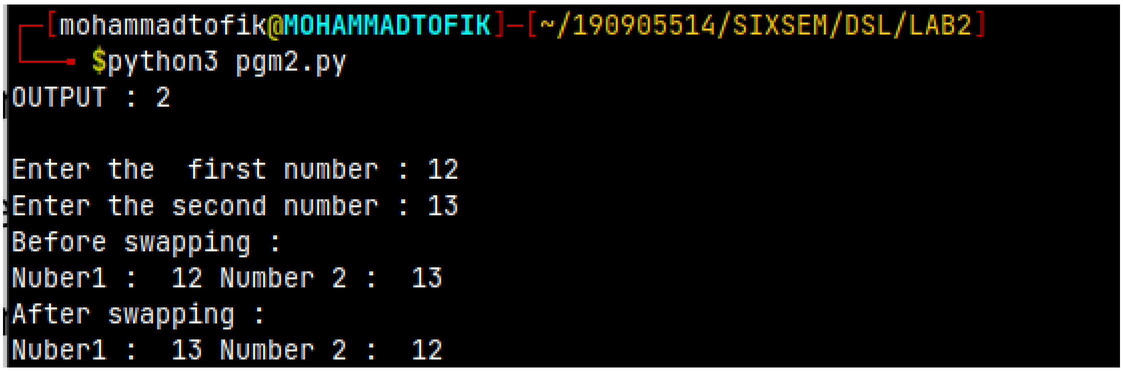
```
import numpy as np

print('OUTPUT : 1\n')
a = int(input('Enter Number: '))
b = int(input('Enter Number: '))
print('Area of rectangle')
print(a*b)
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm1.py
OUTPUT : 1

Enter Number: 5
Enter Number: 6
Area of rectangle
30
```

2.write a program to swap the value of two variable

```
print("OUTPUT : 2\n")
num1 = int(input("Enter the first number : "))
num2 = int(input("Enter the second number : "))
print("Before swapping : ")
print("Nuber1 : ",num1,"Number 2 : ", num2)
temp = num1
num1 = num2
num2 = temp
print("After swapping : ")
print("Nuber1 : ",num1,"Number 2 : ", num2)
```

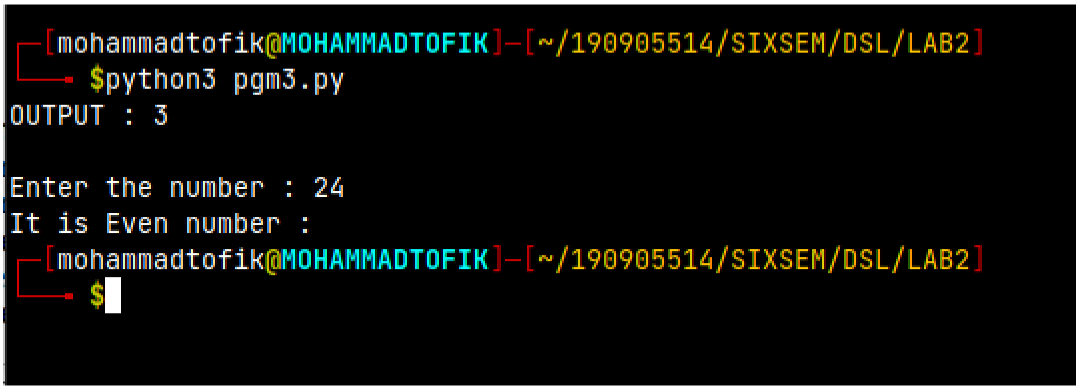


```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm2.py
OUTPUT : 2

Enter the  first number : 12
Enter the second number : 13
Before swapping :
Nuber1 :  12 Number 2 :  13
After swapping :
Nuber1 :  13 Number 2 :  12
```

3.write a program to find whether the given number is odd or even

```
print("OUTPUT : 3\n")
num1 = int(input("Enter the number : "))
if num1 %2 == 0:
```

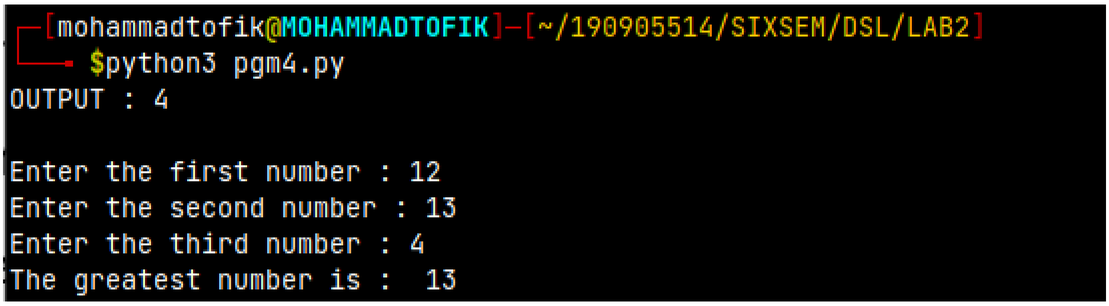
```
print("It is Even number : ")
else:
print("It is odd number : ")
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm3.py
OUTPUT : 3

Enter the number : 24
It is Even number :
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $
```

4.WAP to check largest number among three number

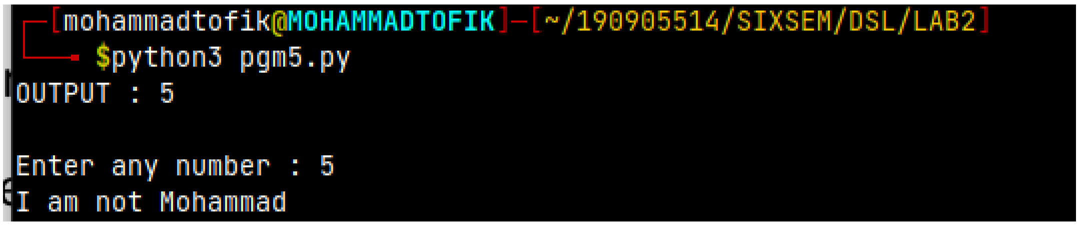
```
print("OUTPUT : 4\n")
num1 = int(input("Enter the first number : "))
num2 = int(input("Enter the second number : "))
num3 = int(input("Enter the third number : "))
if num1 > num2 and num1 > num3:
print("The greatest number is : ", num1)
elif num2 > num3:
print("The greatest number is : ", num2)
else:
print("The greatest number is : ", num3)
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm4.py
OUTPUT : 4

Enter the first number : 12
Enter the second number : 13
Enter the third number : 4
The greatest number is :  13
```

4.WAP to demonstrate while loop with else

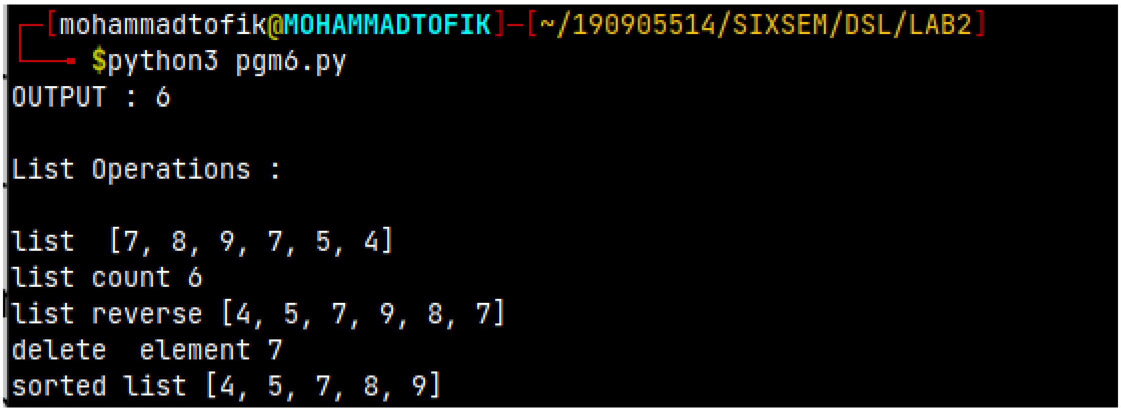
```
print("OUTPUT : 5\n")
num1 = int(input("Enter any number : "))
while num1 > 10:
print("I am Mohammad")
break
else:
print("I am not Mohammad")
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm5.py
OUTPUT : 5

Enter any number : 5
I am not Mohammad
```

6.WAP to demonstrate the list functions and operations

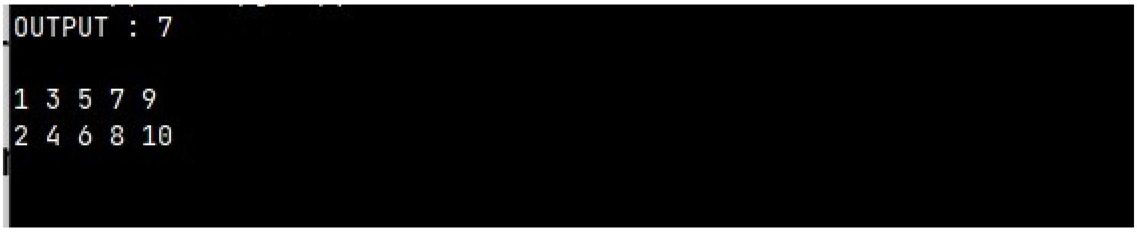
```
print("OUTPUT : 6\n")
print("List Operations : \n")
a = [7,8,9,7,5,4]
print('list ',a)
print('list count',len(a))
a.reverse()
print('list reverse',a)

print('delete element',a.pop())
a.sort()
print('sorted list',a)
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm6.py
OUTPUT : 6

List Operations :

list  [7, 8, 9, 7, 5, 4]
list count 6
list reverse [4, 5, 7, 9, 8, 7]
delete  element 7
sorted list [4, 5, 7, 8, 9]
```

7.consider the tuple(1,3,5,7,9,2,4,6,8,10).WAP to print the half its values in one line and the other half in the next line

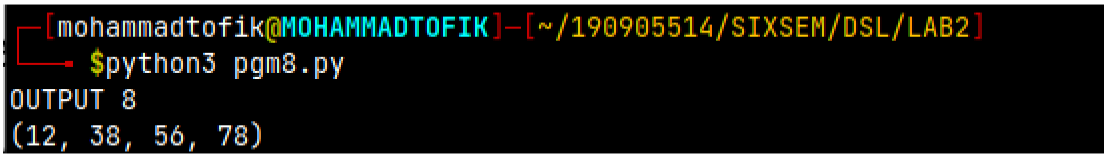
```
print('OUTPUT : 7\n')

tuple1 = (1,3,5,7,9,2,4,6,8,10)
c=0
n=len(tuple1)

for i in tuple1:
c=c+1
if c==int(n/2)+1:
print()
print(i,end=" ")
print("\n\n")
```



```
OUTPUT : 7

1 3 5 7 9
2 4 6 8 10
```

8.consider the tuple(12,7,38,56,78).WAP to print another tuple whose values are even number in the given tuple

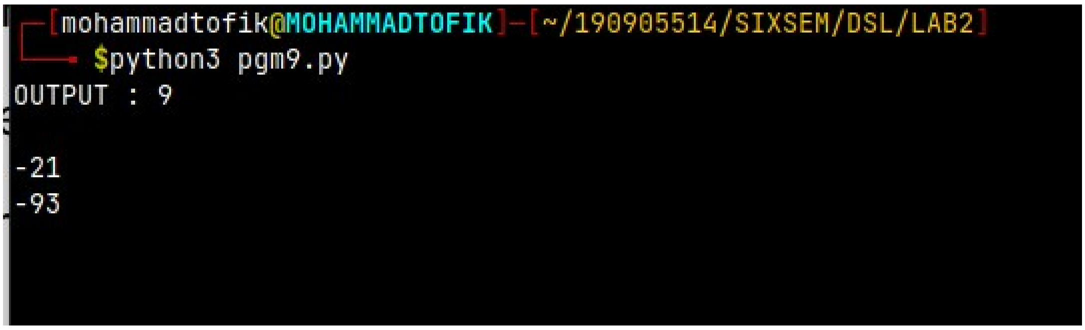
```
print("OUTPUT 8")
tup1 = []
tup =(12, 7, 38, 56, 78)

for num in tup:

if num %2==0:
tup1.append(num)
tup1=tuple(tup1)
print(tup1)
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm8.py
OUTPUT 8
(12, 38, 56, 78)
```

9.WAP to print negative number in a list using for loop Eg.[11,-21,0,45,66,-93]

```
print("OUTPUT : 9\n")
list= [11, -21, 0, 45, 66, -93]
for i in list:
if i<0:
print(i)
print("\n\n")
```

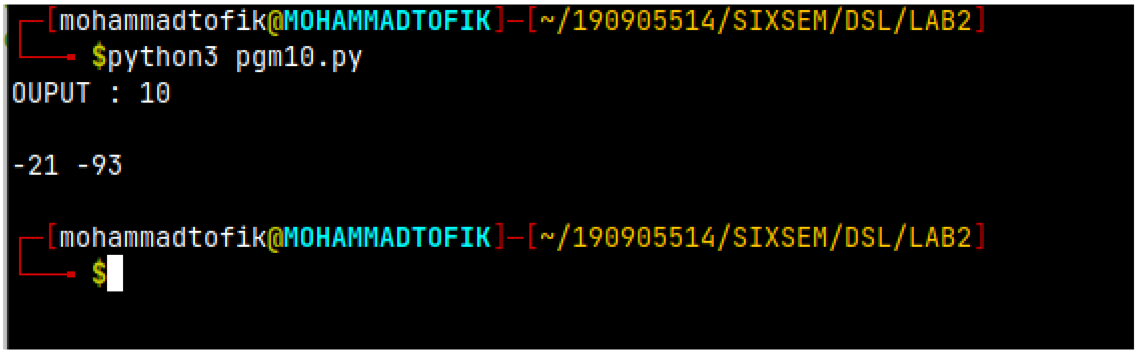


```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm9.py
OUTPUT : 9

-21
-93
```

10.WAP to print negative number in a list using while loop

```
print("OUPUT : 11\n")
list1 = [11, -21, 0, 45, 66, -93]
n=len(list1)
i=0;
while(i<n):
```

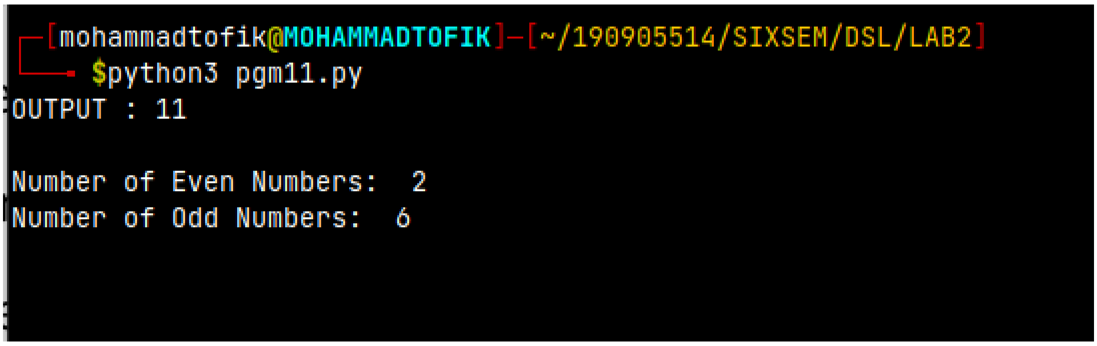
```
if list1[i] < 0:
print(list1[i], end = " ")
i=i+1
print("\n")
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm10.py
OUPUT : 10

-21 -93

┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $
```

11.WAP to to count positive and negative number in a list

```
print("OUTPUT : 11 \n")
list1 = [11, 21, 0, 45, 66, 93]
e=0
o=0;
for num in list1:

if num %2 == 0:
e=e+1;
o=o+1;
print("Number of Even Numbers: ",e)
print("Number of Odd Numbers: ",o)
print("\n\n")
```

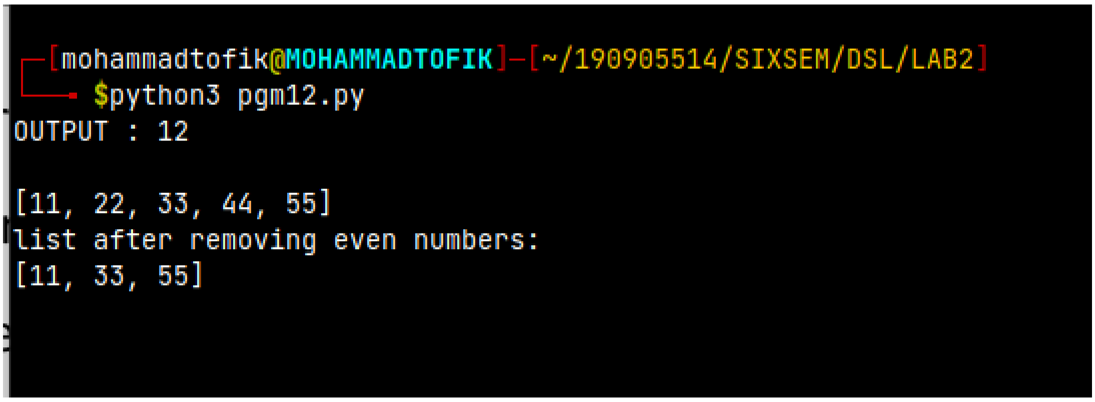


```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm11.py
OUTPUT : 11

Number of Even Numbers:  2
Number of Odd Numbers:  6
```

12.WAP to remove all even elements from the list

```
print("OUTPUT : 12\n")
list = [11, 22, 33, 44, 55]

print (list)

for i in list:
if(i%2 == 0):
```

```
list.remove(i)
print ("list after removing even numbers:")
print (list)
print("\n\n")
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm12.py
OUTPUT : 12

[11, 22, 33, 44, 55]
list after removing even numbers:
[11, 33, 55]
```

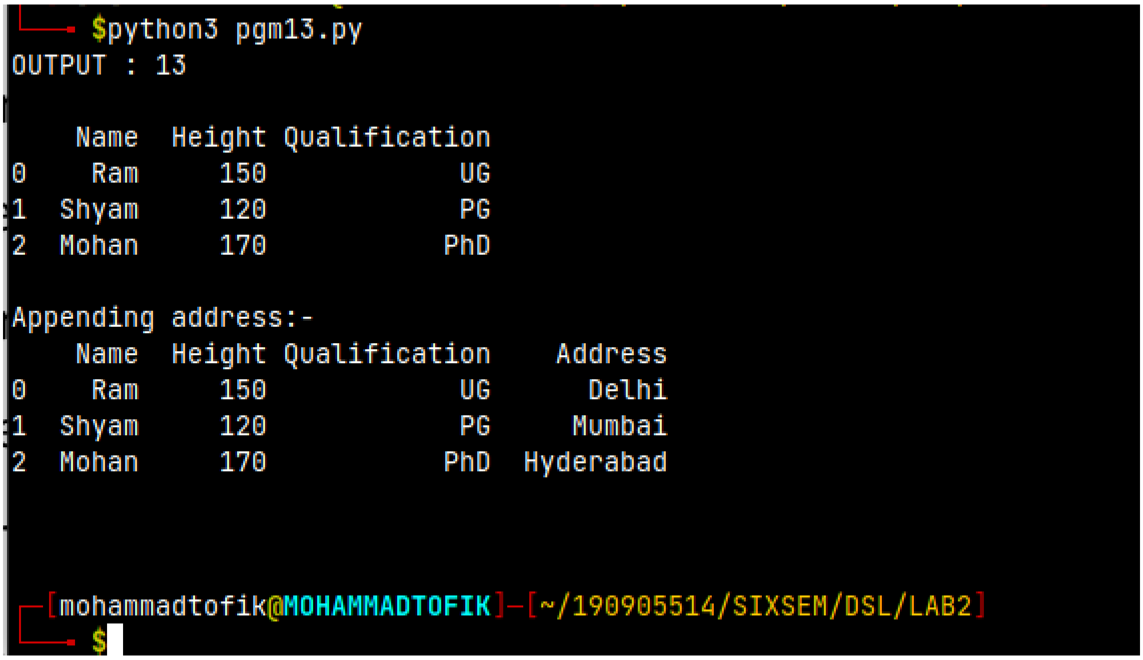
13.Define a dictionary containing Student data {Name, Height, Qualification}

(a)Convert the dictionary into data frame

(b)Declare a list that is to be converted into new column(Address)

(c)Using address as the column name and equate it to the list and display the result.

```
import pandas as pd
print("OUTPUT : 13\n")
dict={"Name":["Ram","Shyam","Mohan"],"Height":[150,120,170],"Qualification":
["UG","PG","PhD"]}
df=pd.DataFrame(dict)
print(df)
address=["Delhi","Mumbai","Hyderabad"]
print("\nAppending address:- ")
df['Address']=address
print(df)
print("\n\n")
```



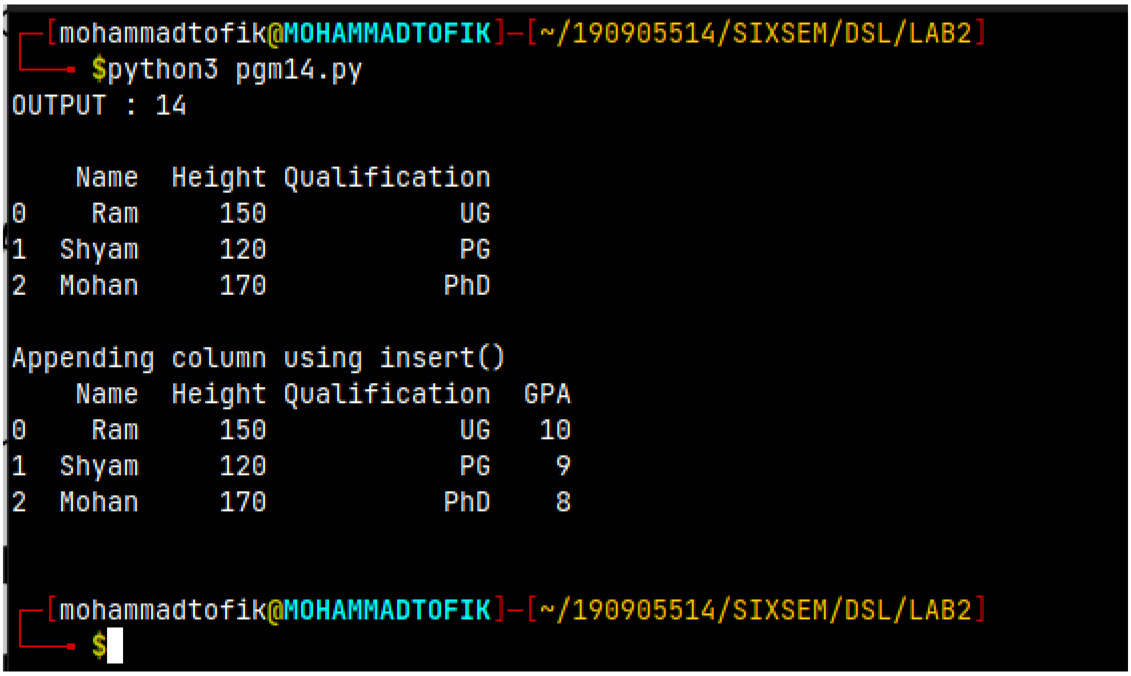
```
└──╼ $python3 pgm13.py
OUTPUT : 13

    Name  Height Qualification
0    Ram     150            UG
1  Shyam     120            PG
2  Mohan     170           PhD

Appending address:-
    Name  Height Qualification    Address
0    Ram     150            UG      Delhi
1  Shyam     120            PG     Mumbai
2  Mohan     170           PhD  Hyderabad



┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $
```

14.Define a dictionary containign student data{Name, Height, Qualification}

(a)Convert the dictionary into DataFrame

(b)Use DataFrame.insert() to add a column and display the result

```
import pandas as pd
print("OUTPUT : 14 \n")
dict={"Name":["Ram","Shyam","Mohan"],"Height":[150,120,170],"Qualification":
["UG","PG","PhD"]}
df=pd.DataFrame(dict)
print(df)
print("\nAppending column using insert()")
gpa=[10,9,8]
df.insert(3, "GPA", gpa, allow_duplicates = False)
print(df)
print("\n")
```



```
┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $python3 pgm14.py
OUTPUT : 14

    Name  Height Qualification
0    Ram     150            UG
1  Shyam     120            PG
2  Mohan     170           PhD

Appending column using insert()
    Name  Height Qualification  GPA
0    Ram     150            UG   10
1  Shyam     120            PG    9
2  Mohan     170           PhD    8


┌─[mohammadtofik@MOHAMMADTOFIK]─[~/190905514/SIXSEM/DSL/LAB2]
└──╼ $
```