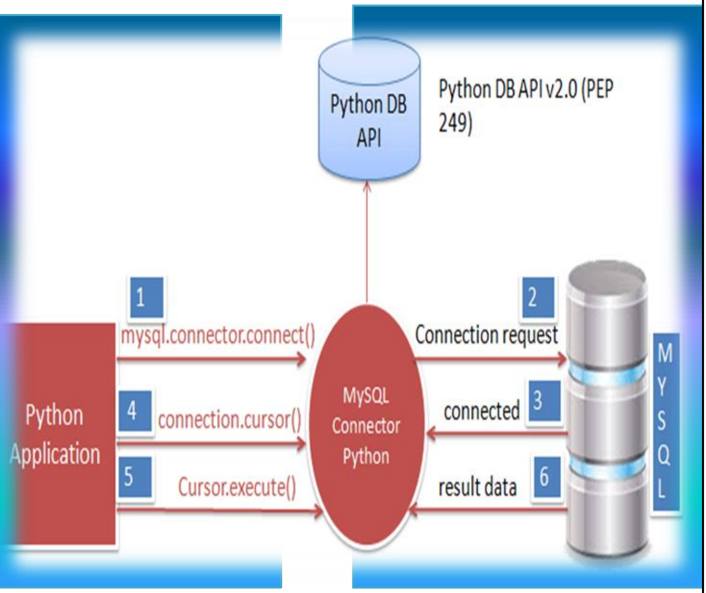
KENDRIYA VIDYALAYA ONGC PANVEL

CLASS XII
INFORMATICS PRACTICES
PRACTICAL FILE

FOR SESSION 2020-21

PREPARED BY:

MRS. SHRUTI SRIVASTAVA, PGT (CS), KV ONGC PANVEL



INDEX

11.	Program to perform select subsets from Dataframe.	09/09/2020	16/09/2020
12.	Program to perform Arithmetic binary operations on dataframe.	16/09/2020	21/09/2020
13.	Program to access and modify the values in Dataframe	21/09/2020	28/09/2020
14.	Program for creating Series objects from Dataframe columns and Dataframe rows.	28/09/2020	07/10/2020
15.	Write a program to create a Dataframe for student details and create a CSV file from it.	07/10/2020	12/10/2020
16.	Write a program to read a CSV file and display contents in different format. Copy the CSV file into other CSV file with different separator.	12/10/2020	16/10/2020
17.	Write a program to draw line chart depicting the prices of the apps and download of the apps.	16/10/2020	19/10/2020
18	Given the school result data, Plot bar graph using function of Dataframe for subject wise analysis of performance of the students.	19/10/2020	22/10/2020
19	Write a program to create a bar chart plotting from the columns of Dataframe.	22/10/2020	29/10/2020
20	Write a program to create different histograms for the weight measurements for 16 small orders of French fries (in grams).	29/10/2020	02/11/2020

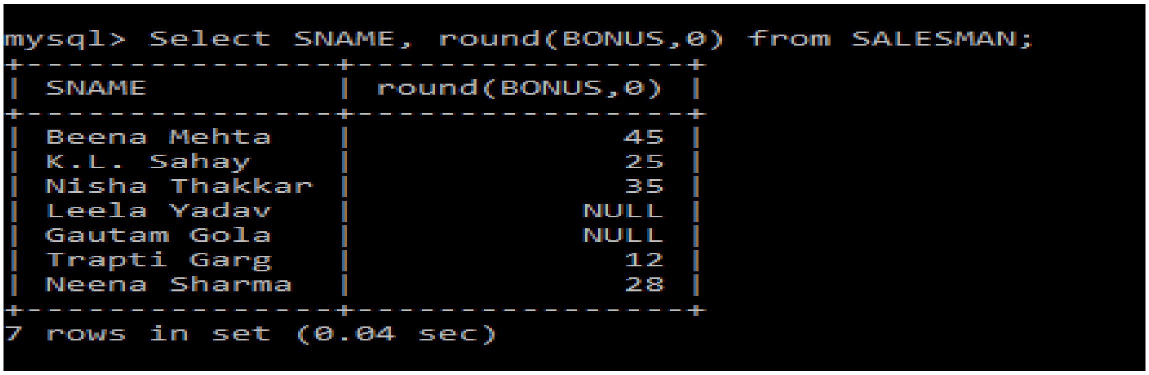
Practical 1-Aim: Database query using SQL (Mathematical, string, Date and time functions in SQL)

Consider table **SALESMAN** with following data:

SNO	SNAME	SALARY	BONUS	DATEOFJOIN
A01	Beena Mehta	30000	45.23	2019-10-29
A02	K. L. Sahay	50000	25.34	2018-03-13
B03	Nisha Thakkar	30000	35.00	2017-03-18
B04	Leela Yadav	80000	NULL	2018-12-31
C05	Gautam Gola	20000	NULL	1989-01-23
C06	Trapti Garg	70000	12.37	1987-06-15
D07	Neena Sharma	50000	27.89	1999-03-18

1) **Display Salesman name , bonus after rounding off to zero decimal places.**

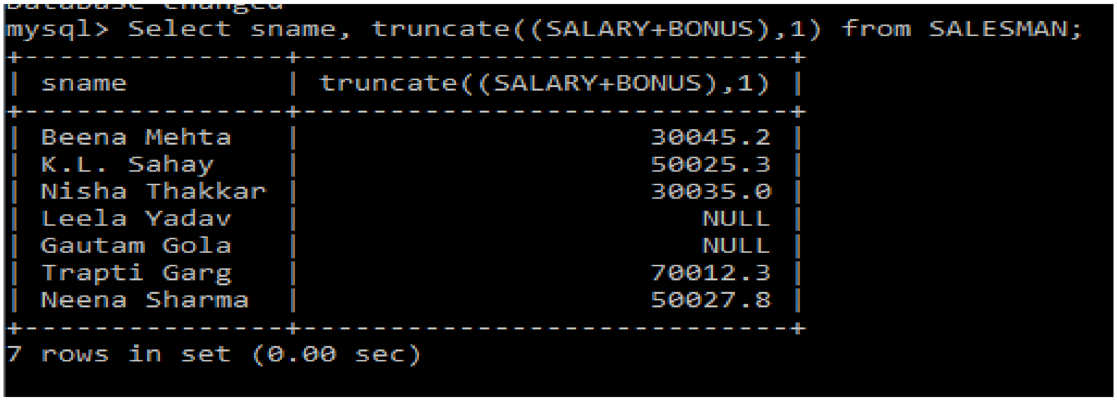
   Select SNAME, round(BONUS,0) from SALESMAN;



```
mysql> Select SNAME, round(BONUS,0) from SALESMAN;
+---------------+----------------+
| SNAME         | round(BONUS,0) |
+---------------+----------------+
| Beena Mehta   |             45 |
| K.L. Sahay    |             25 |
| Nisha Thakkar |             35 |
| Leela Yadav   |           NULL |
| Gautam Gola   |           NULL |
| Trapti Garg   |             12 |
| Neena Sharma  |             28 |
+---------------+----------------+
7 rows in set (0.04 sec)
```

2) **Display name, total salary of all salesman after addition of salary and bonus and truncate it to 1 decimal places.**

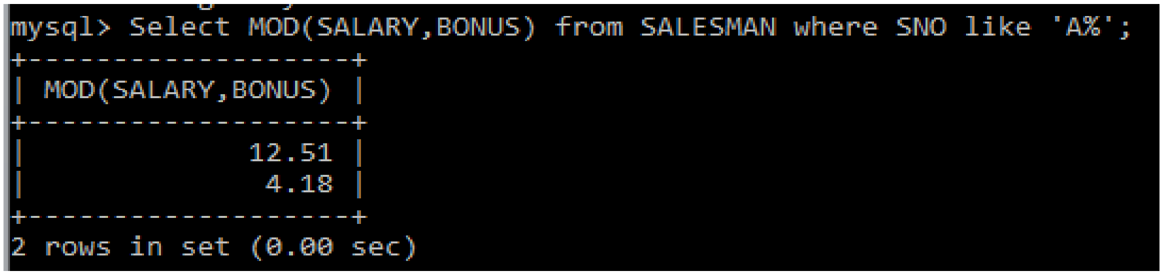
   Select sname, truncate((SALARY+BONUS),1) from SALESMAN;



```
Database changed
mysql> Select sname, truncate((SALARY+BONUS),1) from SALESMAN;
+---------------+----------------------------+
| sname         | truncate((SALARY+BONUS),1) |
+---------------+----------------------------+
| Beena Mehta   |                    30045.2 |
| K.L. Sahay    |                    50025.3 |
| Nisha Thakkar |                    30035.0 |
| Leela Yadav   |                       NULL |
| Gautam Gola   |                       NULL |
| Trapti Garg   |                    70012.3 |
| Neena Sharma  |                    50027.8 |
+---------------+----------------------------+
7 rows in set (0.00 sec)
```

3) Display remainder of salary and bonus of Salesman whose SNO starting with ‘A’

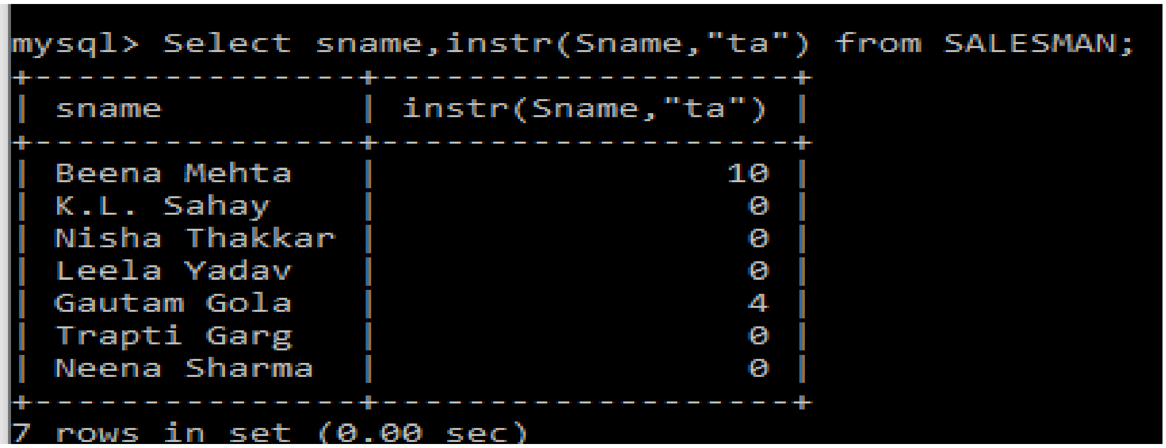
Select MOD(SALARY,BONUS) from SALESMAN where SNO like ’A%’;



```
mysql> Select MOD(SALARY,BONUS) from SALESMAN where SNO like 'A%';
+-------------------+
| MOD(SALARY,BONUS) |
+-------------------+
|             12.51 |
|              4.18 |
+-------------------+
2 rows in set (0.00 sec)
```

4) Display position of occurrence of string “ta” in salesmen name.

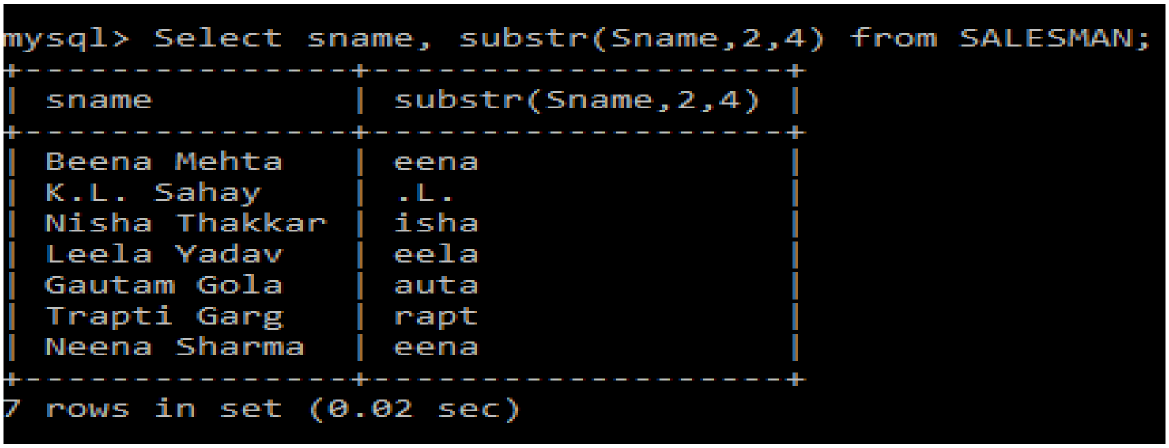
Select sname, instr(Sname,”ta”) from SALESMAN;



```
mysql> Select sname,instr(Sname,"ta") from SALESMAN;
+---------------+-------------------+
| sname         | instr(Sname,"ta") |
+---------------+-------------------+
| Beena Mehta   |                10 |
| K.L. Sahay    |                 0 |
| Nisha Thakkar |                 0 |
| Leela Yadav   |                 0 |
| Gautam Gola   |                 4 |
| Trapti Garg   |                 0 |
| Neena Sharma  |                 0 |
+---------------+-------------------+
7 rows in set (0.00 sec)
```

5) Display four characters from salesman name starting from second character.

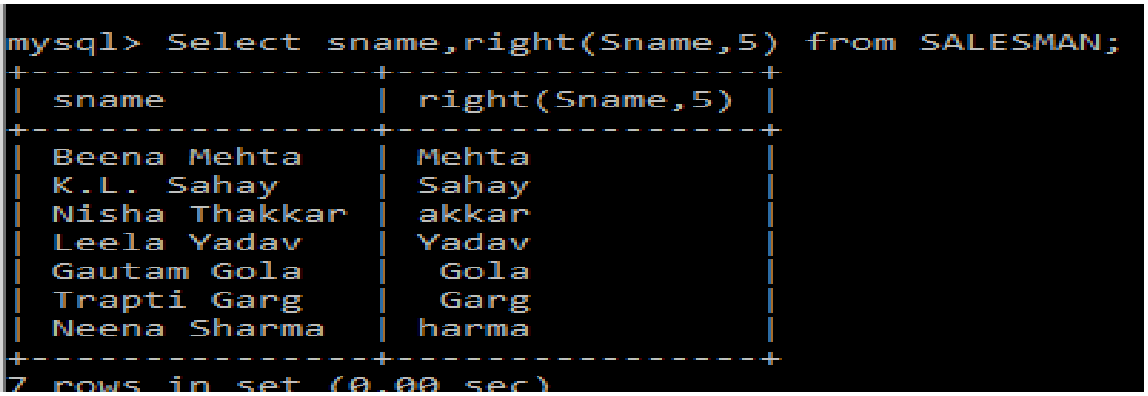
Select sname, substr(Sname,2,4) from SALESMAN;



```
mysql> Select sname, substr(Sname,2,4) from SALESMAN;
+---------------+-------------------+
| sname         | substr(Sname,2,4) |
+---------------+-------------------+
| Beena Mehta   | eena              |
| K.L. Sahay    | .L.               |
| Nisha Thakkar | isha              |
| Leela Yadav   | eela              |
| Gautam Gola   | auta              |
| Trapti Garg   | rapt              |
| Neena Sharma  | eena              |
+---------------+-------------------+
7 rows in set (0.02 sec)
```

6) **Display last 5 characters of name of SALESMAN**.

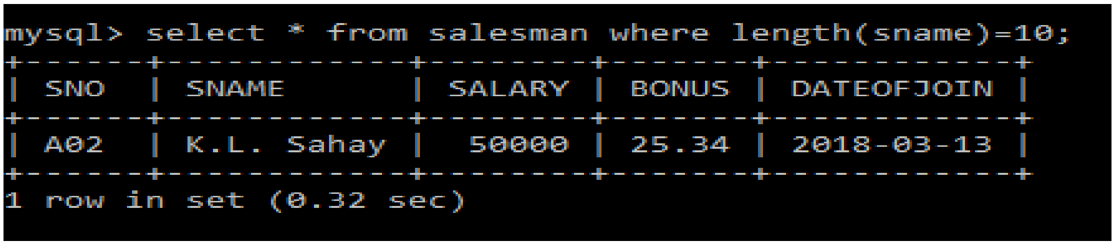
Select sname, right(Sname,5) from SALESMAN;



```
mysql> Select sname,right(Sname,5) from SALESMAN;
+----------------+----------------+
| sname          | right(Sname,5) |
+----------------+----------------+
| Beena Mehta    | Mehta          |
| K.L. Sahay     | Sahay          |
| Nisha Thakkar  | akkar          |
| Leela Yadav    | Yadav          |
| Gautam Gola    |  Gola          |
| Trapti Garg    |  Garg          |
| Neena Sharma   | harma          |
+----------------+----------------+
7 rows in set (0.00 sec)
```

7) **Display details of salesman whose name containing 10 characters.**

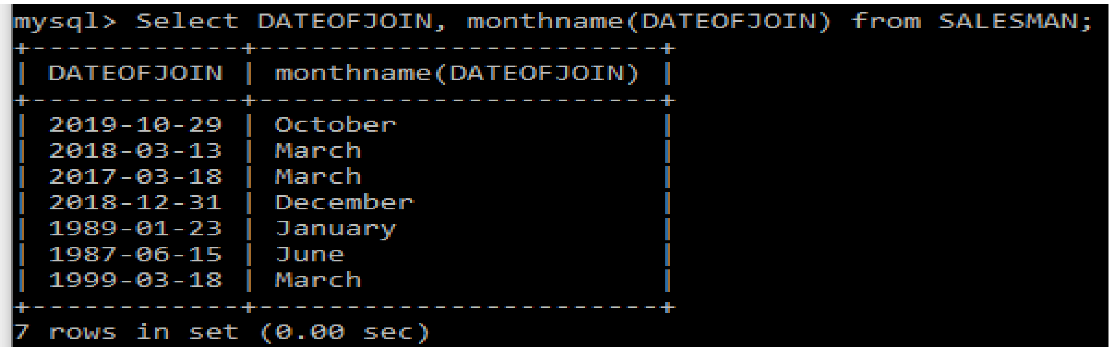
Select * from salesman where length(sname)=10;



```
mysql> select * from salesman where length(sname)=10;
+------+------------+--------+-------+------------+
| SNO  | SNAME      | SALARY | BONUS | DATEOFJOIN |
+------+------------+--------+-------+------------+
| A02  | K.L. Sahay |  50000 | 25.34 | 2018-03-13 |
+------+------------+--------+-------+------------+
1 row in set (0.32 sec)
```

8) Display month name for the date of join of salesman

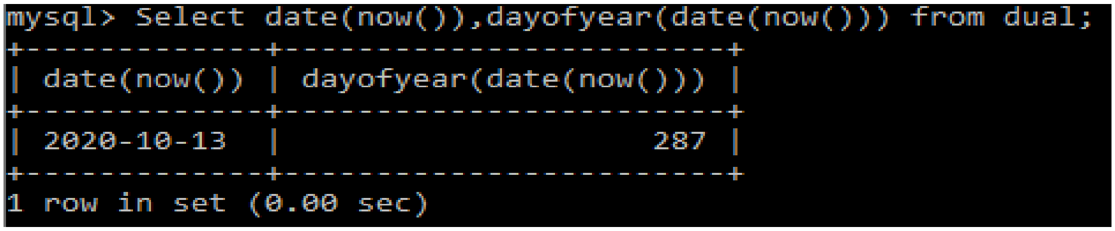
Select DATEOFJOIN, monthname(DATEOFJOIN) from SALESMAN;



```
mysql> Select DATEOFJOIN, monthname(DATEOFJOIN) from SALESMAN;
+------------+-----------------------+
| DATEOFJOIN | monthname(DATEOFJOIN) |
+------------+-----------------------+
| 2019-10-29 | October               |
| 2018-03-13 | March                 |
| 2017-03-18 | March                 |
| 2018-12-31 | December              |
| 1989-01-23 | January               |
| 1987-06-15 | June                  |
| 1999-03-18 | March                 |
+------------+-----------------------+
7 rows in set (0.00 sec)
```

9) Display currentdate and day of the year of current date.

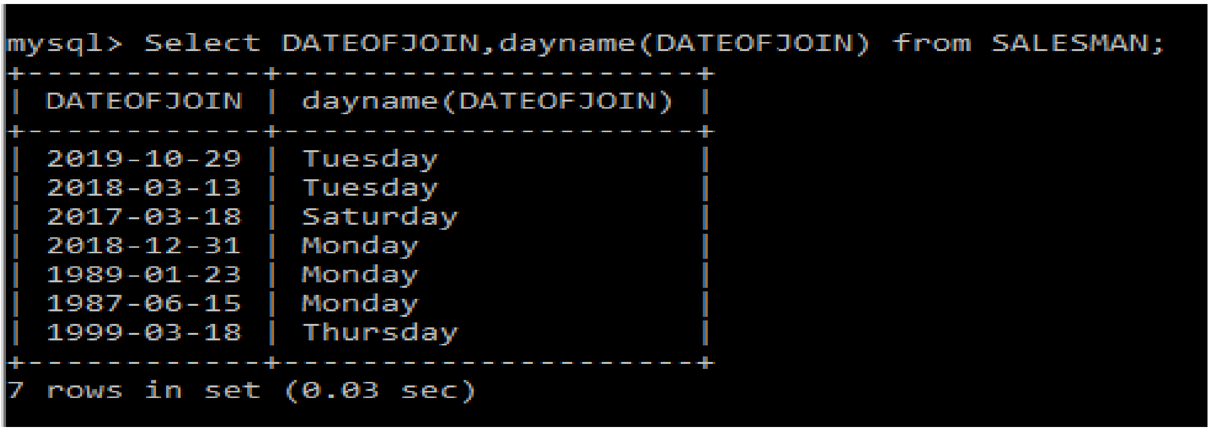
Select date (now()),dayofyear(date(now())) from dual;



```
mysql> Select date(now()),dayofyear(date(now())) from dual;
+-------------+------------------------+
| date(now()) | dayofyear(date(now())) |
+-------------+------------------------+
| 2020-10-13  |                    287 |
+-------------+------------------------+
1 row in set (0.00 sec)
```

10) Display name of the weekday for the DATEOFJOIN of SALESMAN;

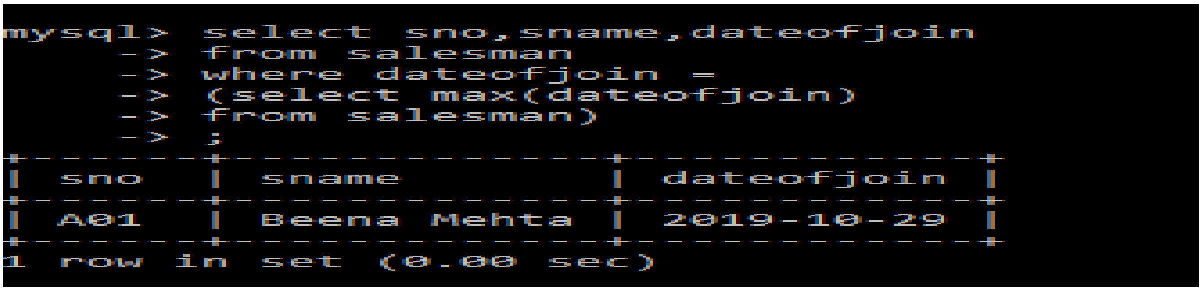
Select DATEOFJOIN,dayname(DATEOFJOIN) from SALESMAN;



```
mysql> Select DATEOFJOIN,dayname(DATEOFJOIN) from SALESMAN;
+------------+---------------------+
| DATEOFJOIN | dayname(DATEOFJOIN) |
+------------+---------------------+
| 2019-10-29 | Tuesday             |
| 2018-03-13 | Tuesday             |
| 2017-03-18 | Saturday            |
| 2018-12-31 | Monday              |
| 1989-01-23 | Monday              |
| 1987-06-15 | Monday              |
| 1999-03-18 | Thursday            |
+------------+---------------------+
7 rows in set (0.03 sec)
```

11) Display SNO, name of the youngest SALESMAN.

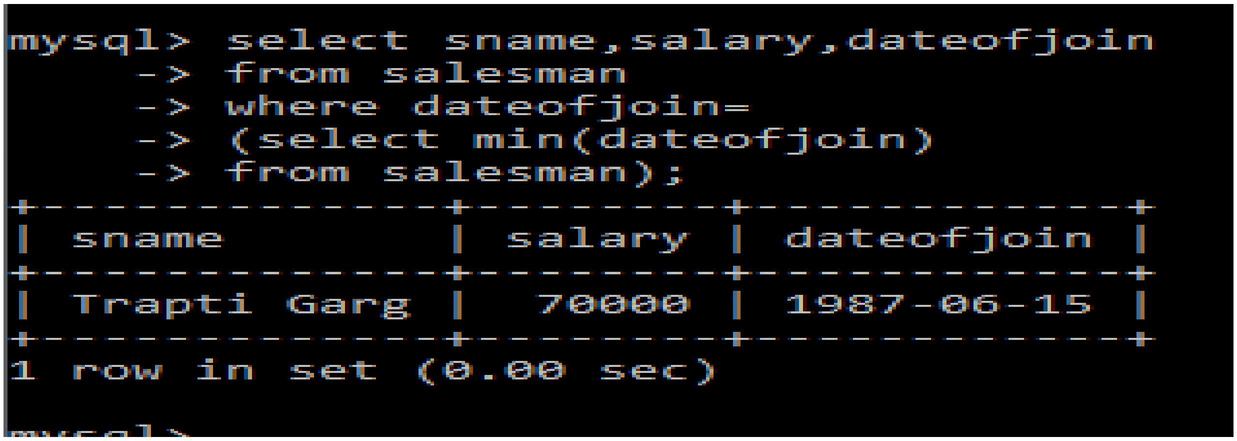
Select sno, sname, dateofjoin from salesman where dateofjoin=(select max(DATEOFJOIN) from SALESMAN);



```
mysql> select sno,sname,dateofjoin
    -> from salesman
    -> where dateofjoin =
    -> (select max(dateofjoin)
    -> from salesman)
    -> ;
+------+---------------+------------+
| sno  | sname         | dateofjoin |
+------+---------------+------------+
| A01  | Beena Mehta   | 2019-10-29 |
+------+---------------+------------+
1 row in set (0.00 sec)
```

12) Display name and salary of the oldest SALESMAN.

Select sname, salary, dateofjoin from salesman where dateofjoin=(select min(dateofjoin) from salesman);



```
mysql> select sname,salary,dateofjoin
    -> from salesman
    -> where dateofjoin=
    -> (select min(dateofjoin)
    -> from salesman);
+--------------+--------+------------+
| sname        | salary | dateofjoin |
+--------------+--------+------------+
| Trapti Garg  |  70000 | 1987-06-15 |
+--------------+--------+------------+
1 row in set (0.00 sec)

mysql>
```

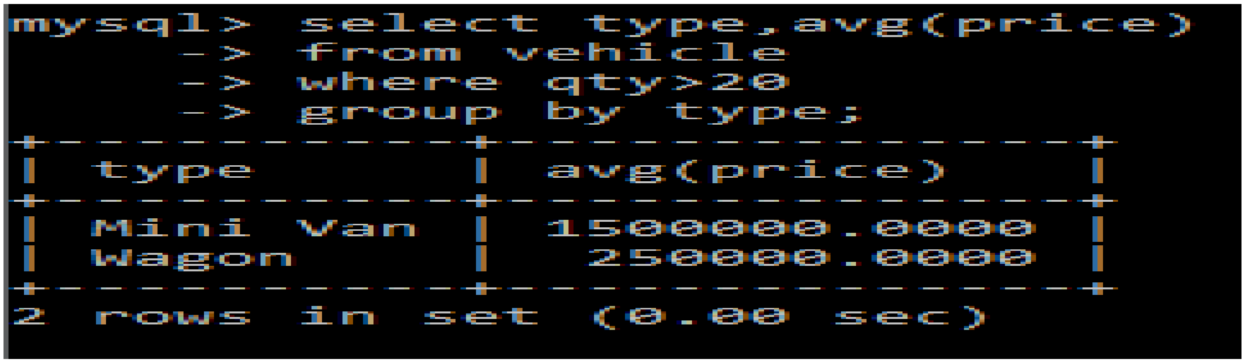
Practical 2-Aim: Database query using SQL (Aggregate functions ,Group by ,order by query in SQL)

Consider the following table **Vehicle**:

V_no	Type	Company	Price	Qty
AW125	Wagon	Maruti	250000	25
J00083	Jeep	Mahindra	4000000	15
S9090	SUV	Mistubishi	2500000	18
M0892	Mini Van	Datsun	1500000	26
W9760	SUV	Maruti	2500000	18
R2409	Mini Van	Mahindra	350000	15

1. **Display the average price of each type of vehicle having quantity more than 20.**

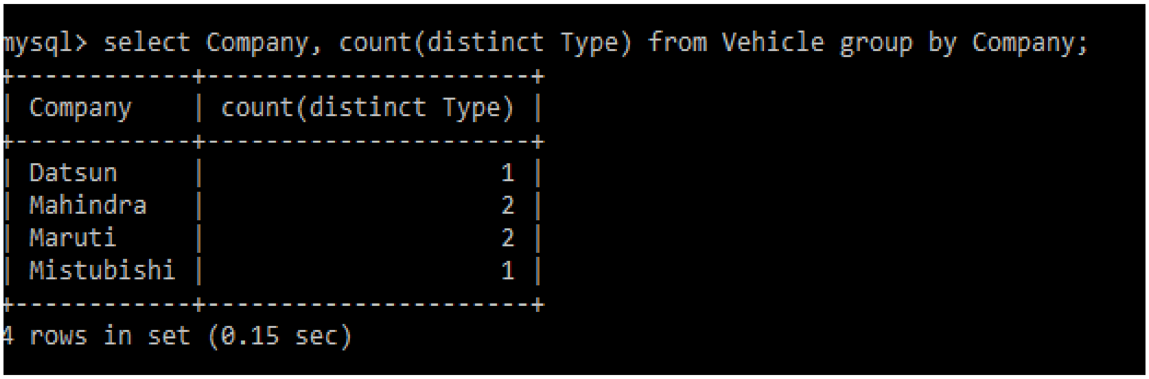
Select Type, avg(price) from vehicle where qty>20 group by Type;



```
mysql> select type,avg(price)
    -> from vehicle
    -> where qty>20
    -> group by type;
+-----------+------------------+
| type      | avg(price)       |
+-----------+------------------+
| Mini Van  | 1500000.0000     |
| Wagon     |  250000.0000     |
+-----------+------------------+
2 rows in set (0.00 sec)
```

2. **Count the type of vehicles manufactured by each company.**

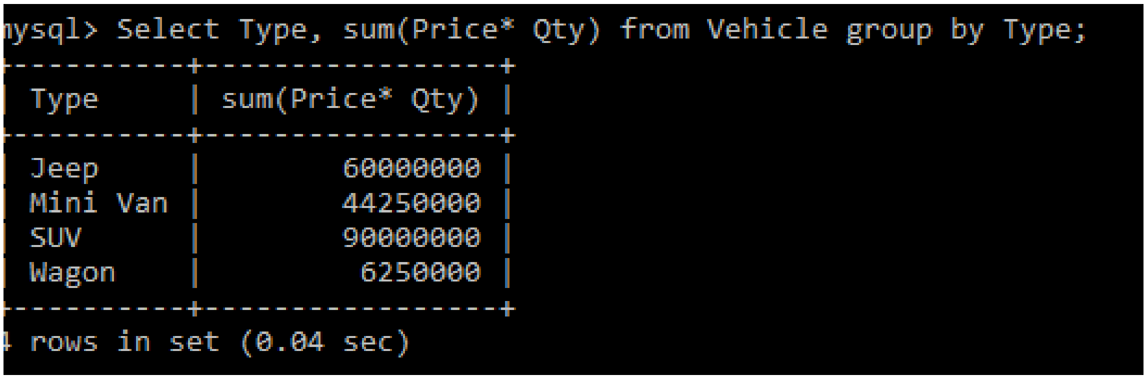
Select Company, count(distinct Type) from Vehicle group by Company;



```
mysql> select Company, count(distinct Type) from Vehicle group by Company;
+------------+----------------------+
| Company    | count(distinct Type) |
+------------+----------------------+
| Datsun     |                    1 |
| Mahindra   |                    2 |
| Maruti     |                    2 |
| Mistubishi |                    1 |
+------------+----------------------+
4 rows in set (0.15 sec)
```

3. Display total price of all types of vehicle.

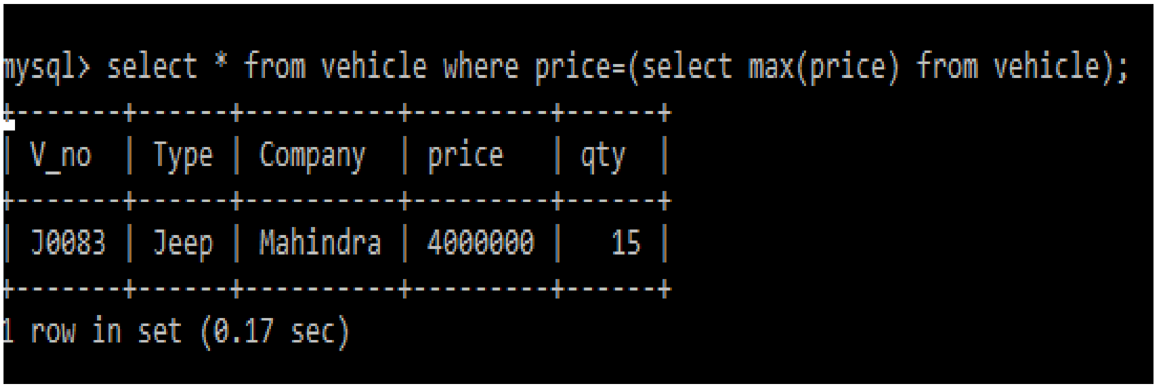
Select Type, sum(Price* Qty) from Vehicle group by Type;



```
mysql> Select Type, sum(Price* Qty) from Vehicle group by Type;
+----------+-----------------+
| Type     | sum(Price* Qty) |
+----------+-----------------+
| Jeep     |        60000000 |
| Mini Van |        44250000 |
| SUV      |        90000000 |
| Wagon    |         6250000 |
+----------+-----------------+
4 rows in set (0.04 sec)
```

4. Display the details of the vehichle having maximum price.

Select * from vehicle where price=(select max(price) from vehicle);



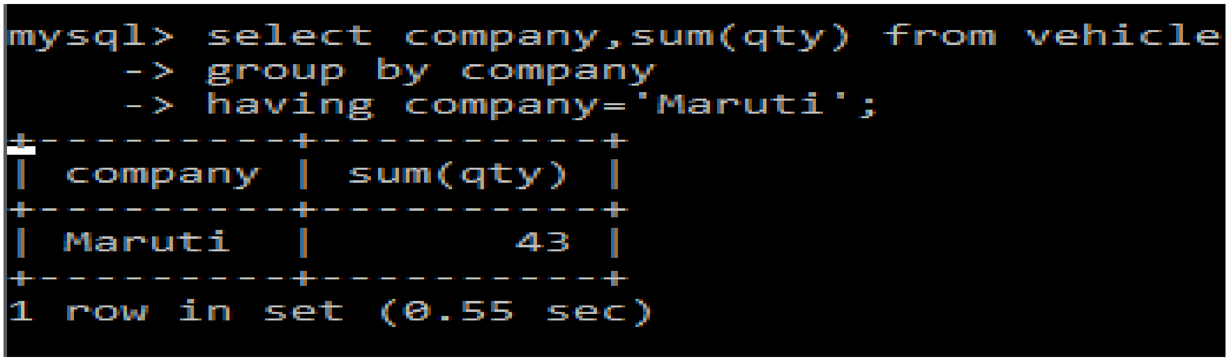
```
mysql> select * from vehicle where price=(select max(price) from vehicle);
+-------+------+----------+---------+------+
| V_no  | Type | Company  | price   | qty  |
+-------+------+----------+---------+------+
| J0083 | Jeep | Mahindra | 4000000 |   15 |
+-------+------+----------+---------+------+
1 row in set (0.17 sec)
```

5. Display total vehicles of Maruti company.

Select company,sum(qty) from vehicle

group by company

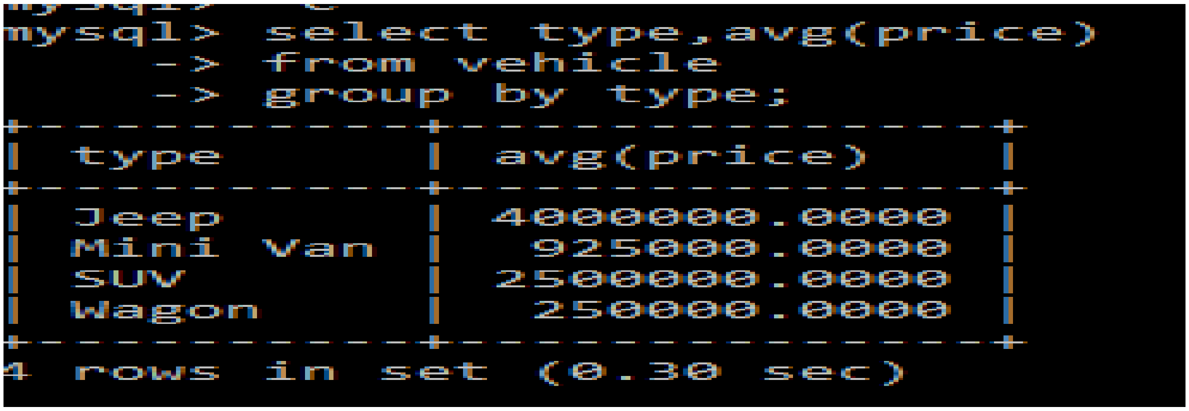
having company='Maruti';



```
mysql> select company,sum(qty) from vehicle
    -> group by company
    -> having company='Maruti';
+---------+----------+
| company | sum(qty) |
+---------+----------+
| Maruti  |       43 |
+---------+----------+
1 row in set (0.55 sec)
```

6. Display average price of all type of vehicles.

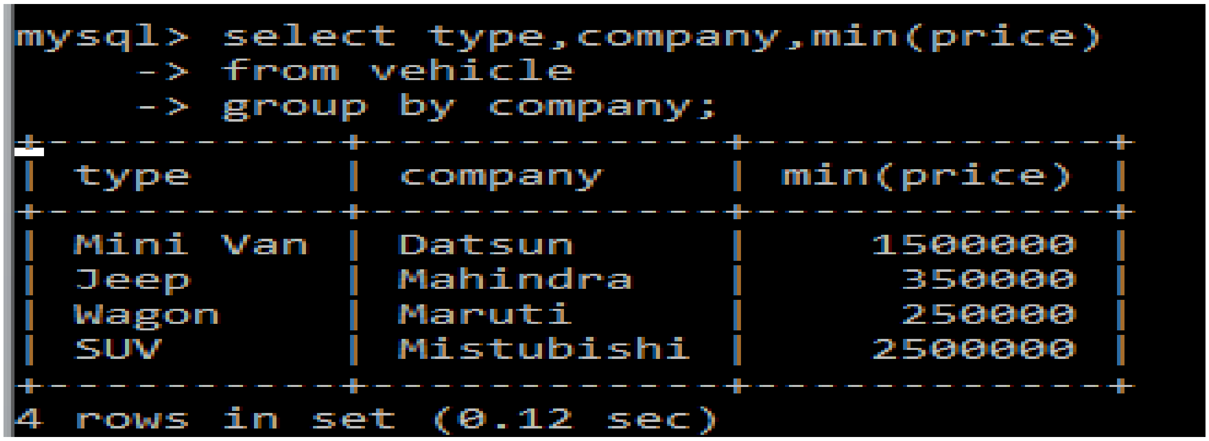
Select type,avg(price) from vehicle group by type;



```
mysql> select type,avg(price)
    -> from vehicle
    -> group by type;
+-----------+----------------+
| type      | avg(price)     |
+-----------+----------------+
| Jeep      | 4000000.0000   |
| Mini Van  |  925000.0000   |
| SUV       | 2500000.0000   |
| Wagon     |  250000.0000   |
+-----------+----------------+
4 rows in set (0.30 sec)
```

7. Display type and minimum price of each vehicle company.

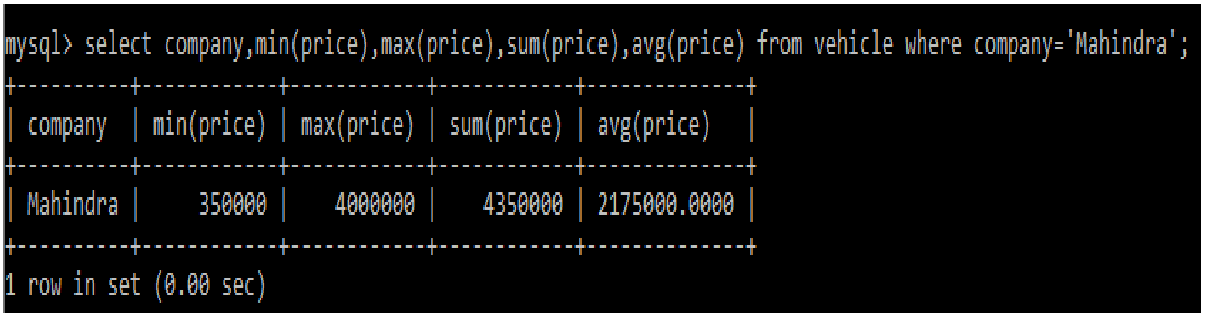
Select type,company,min(price) from vehicle group by company;



```
mysql> select type,company,min(price)
    -> from vehicle
    -> group by company;
+----------+------------+------------+
| type     | company    | min(price) |
+----------+------------+------------+
| Mini Van | Datsun     |    1500000 |
| Jeep     | Mahindra   |     350000 |
| Wagon    | Maruti     |     250000 |
| SUV      | Mistubishi |    2500000 |
+----------+------------+------------+
4 rows in set (0.12 sec)
```

8. Display minimum, maximum, total and average price of Mahindra company vehicles.

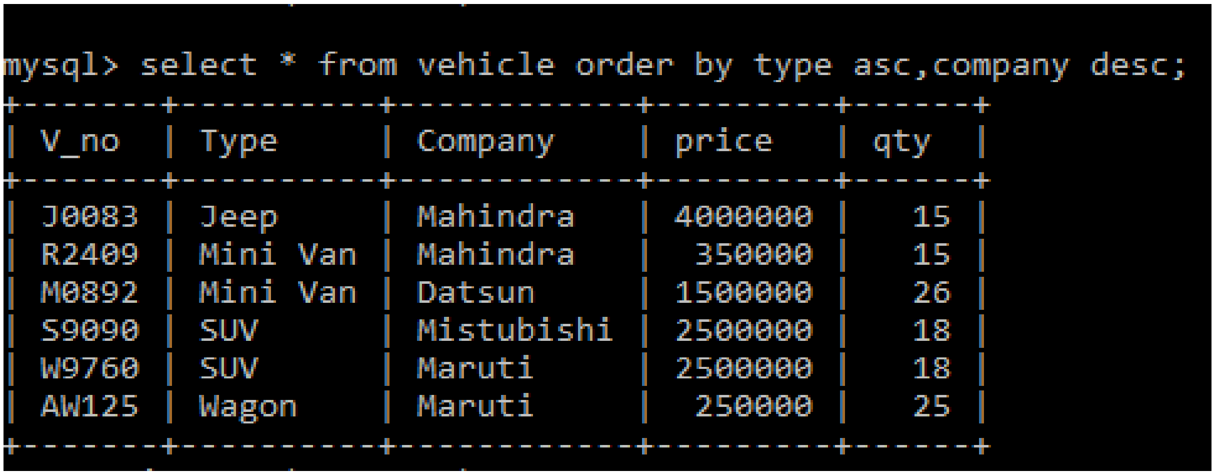
Select company, min(price),max(price),sum(price),avg(price) from vehicle where company='Mahindra';



```
mysql> select company,min(price),max(price),sum(price),avg(price) from vehicle where company='Mahindra';
+----------+------------+------------+------------+--------------+
| company  | min(price) | max(price) | sum(price) | avg(price)   |
+----------+------------+------------+------------+--------------+
| Mahindra |     350000 |    4000000 |    4350000 | 2175000.0000 |
+----------+------------+------------+------------+--------------+
1 row in set (0.00 sec)
```

9. **Display details of all vehicles in ascending order of their price**.

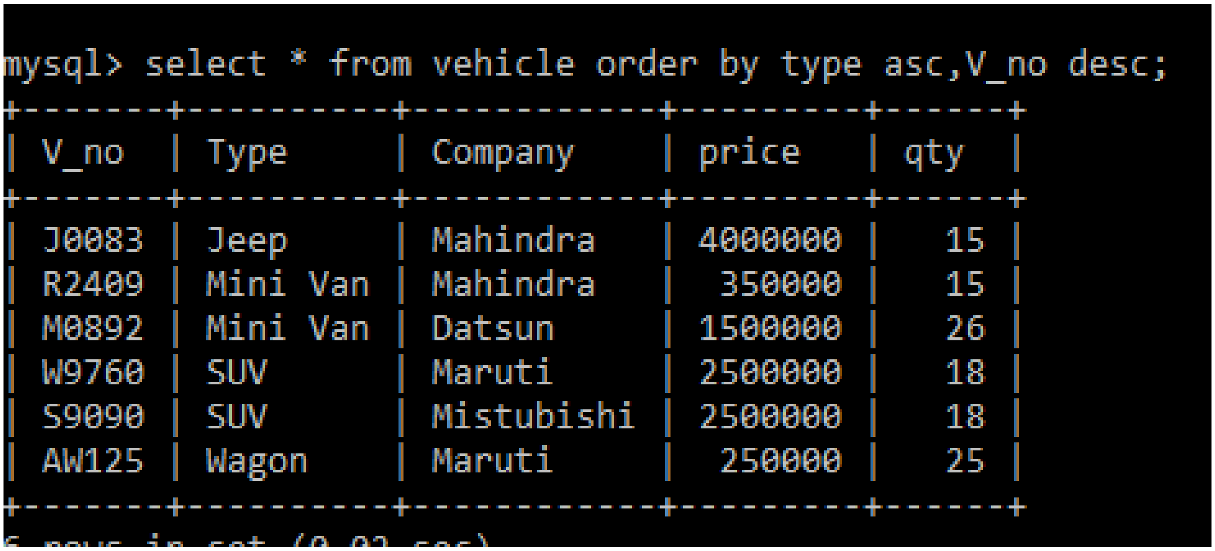
Select * from vehicle order by price asc;



```
mysql> select * from vehicle order by type asc,company desc;
+-------+----------+------------+---------+------+
| V_no  | Type     | Company    | price   | qty  |
+-------+----------+------------+---------+------+
| J0083 | Jeep     | Mahindra   | 4000000 |   15 |
| R2409 | Mini Van | Mahindra   |  350000 |   15 |
| M0892 | Mini Van | Datsun     | 1500000 |   26 |
| S9090 | SUV      | Mistubishi | 2500000 |   18 |
| W9760 | SUV      | Maruti     | 2500000 |   18 |
| AW125 | Wagon    | Maruti     |  250000 |   25 |
+-------+----------+------------+---------+------+
```

10.Display details of all vehicles in ascending order of type and descending order of vehicle number.

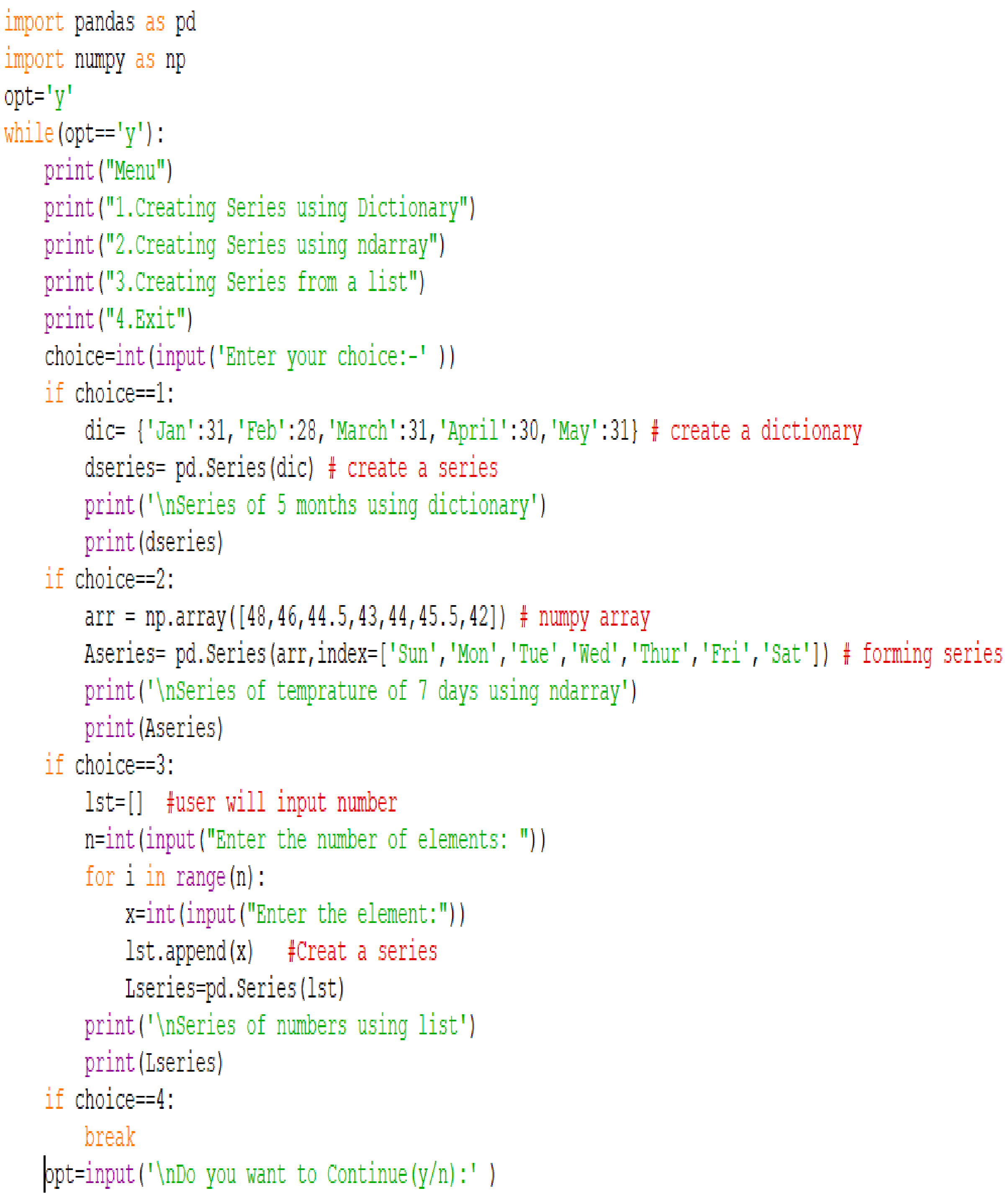
Select * from vehicle order by type asc,v_no desc;



```
mysql> select * from vehicle order by type asc,V_no desc;
+-------+----------+------------+---------+------+
| V_no  | Type     | Company    | price   | qty  |
+-------+----------+------------+---------+------+
| J0083 | Jeep     | Mahindra   | 4000000 |   15 |
| R2409 | Mini Van | Mahindra   |  350000 |   15 |
| M0892 | Mini Van | Datsun     | 1500000 |   26 |
| W9760 | SUV      | Maruti     | 2500000 |   18 |
| S9090 | SUV      | Mistubishi | 2500000 |   18 |
| AW125 | Wagon    | Maruti     |  250000 |   25 |
+-------+----------+------------+---------+------+
6 rows in set (0.02 sec)
```

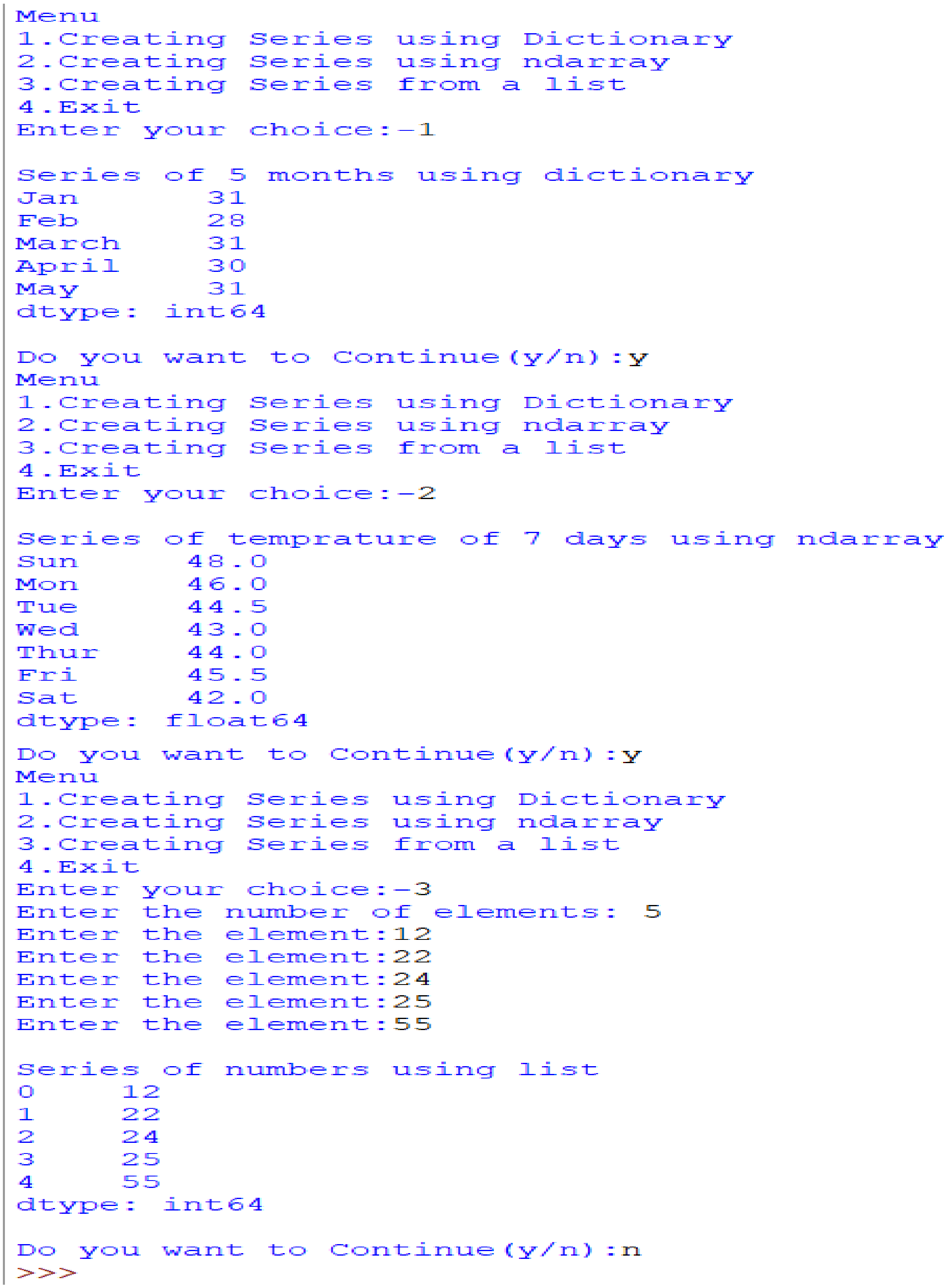
Practical 3-Write a menu driven program to create a pandas series from a dictionary, ndarray and list.

Solution:



```
import pandas as pd
import numpy as np
opt='y'
while(opt=='y'):
    print("Menu")
    print("1.Creating Series using Dictionary")
    print("2.Creating Series using ndarray")
    print("3.Creating Series from a list")
    print("4.Exit")
    choice=int(input('Enter your choice:-' ))
    if choice==1:
        dic= {'Jan':31,'Feb':28,'March':31,'April':30,'May':31} # create a dictionary
        dseries= pd.Series(dic) # create a series
        print('\nSeries of 5 months using dictionary')
        print(dseries)
    if choice==2:
        arr = np.array([48,46,44.5,43,44,45.5,42]) # numpy array
        Aseries= pd.Series(arr,index=['Sun','Mon','Tue','Wed','Thur','Fri','Sat']) # forming series
        print('\nSeries of temprature of 7 days using ndarray')
        print(Aseries)
    if choice==3:
        lst=[]  #user will input number
        n=int(input("Enter the number of elements: "))
        for i in range(n):
            x=int(input("Enter the element:"))
            lst.append(x)   #Creat a series
            Lseries=pd.Series(lst)
        print('\nSeries of numbers using list')
        print(Lseries)
    if choice==4:
        break
    opt=input('\nDo you want to Continue(y/n):' )
```

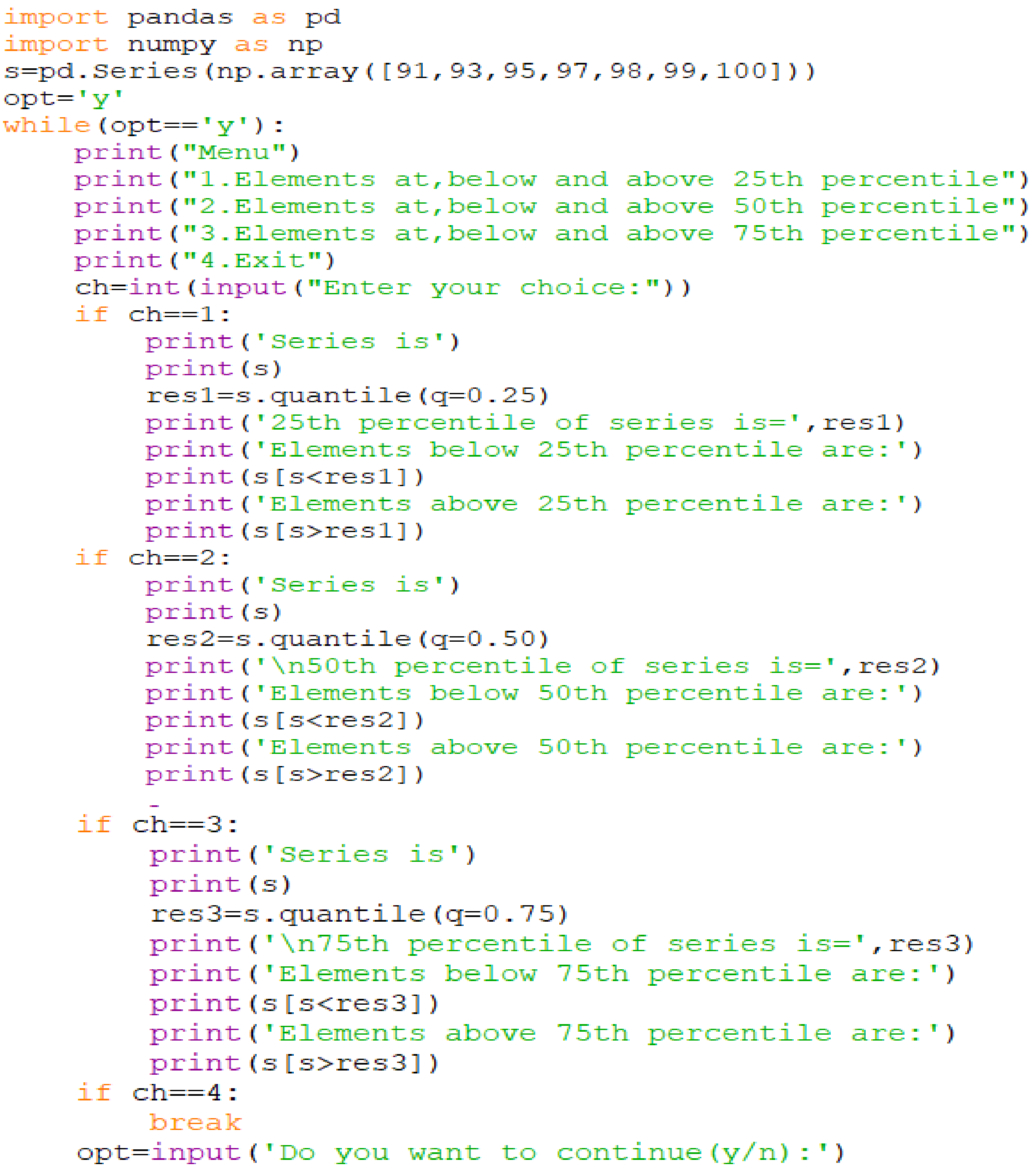
Output:



```
Menu
1.Creating Series using Dictionary
2.Creating Series using ndarray
3.Creating Series from a list
4.Exit
Enter your choice:-1

Series of 5 months using dictionary
Jan       31
Feb       28
March     31
April     30
May       31
dtype: int64

Do you want to Continue(y/n):y
Menu
1.Creating Series using Dictionary
2.Creating Series using ndarray
3.Creating Series from a list
4.Exit
Enter your choice:-2

Series of temprature of 7 days using ndarray
Sun      48.0
Mon      46.0
Tue      44.5
Wed      43.0
Thur     44.0
Fri      45.5
Sat      42.0
dtype: float64

Do you want to Continue(y/n):y
Menu
1.Creating Series using Dictionary
2.Creating Series using ndarray
3.Creating Series from a list
4.Exit
Enter your choice:-3
Enter the number of elements: 5
Enter the element:12
Enter the element:22
Enter the element:24
Enter the element:25
Enter the element:55

Series of numbers using list
0     12
1     22
2     24
3     25
4     55
dtype: int64

Do you want to Continue(y/n):n
>>>
```

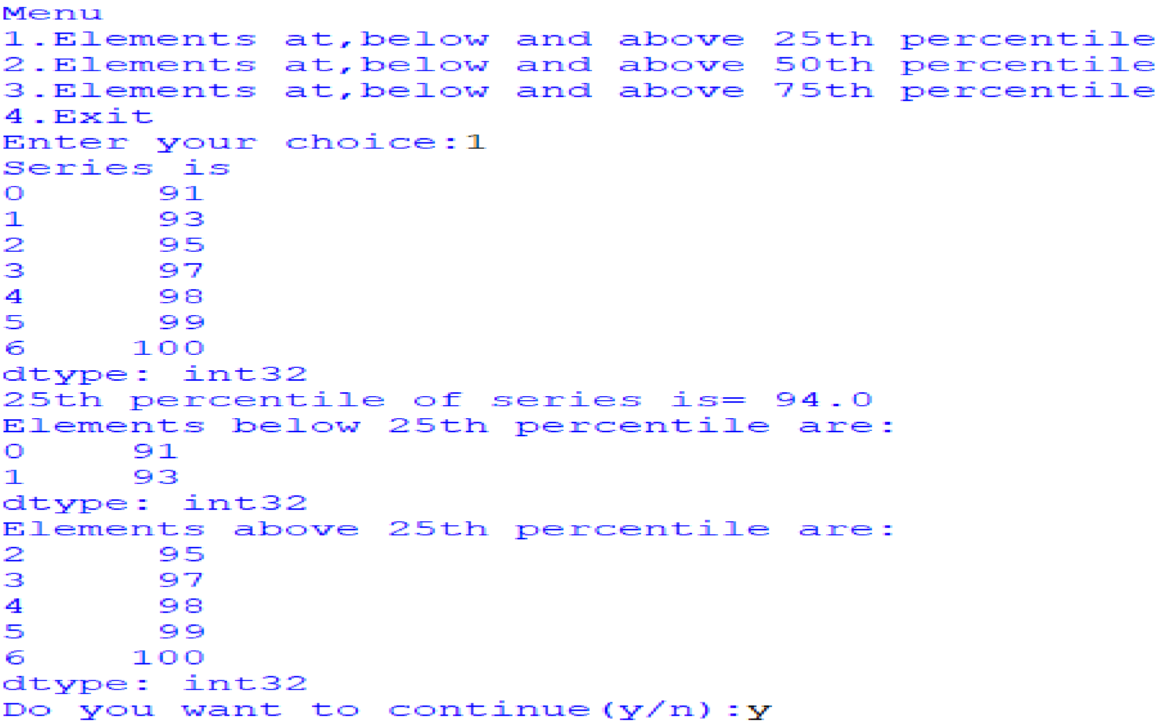
Practical 4-Write a menu driven program to print all the elements that are at, above & below 25th , 50th and 75th percentile.

Solution:

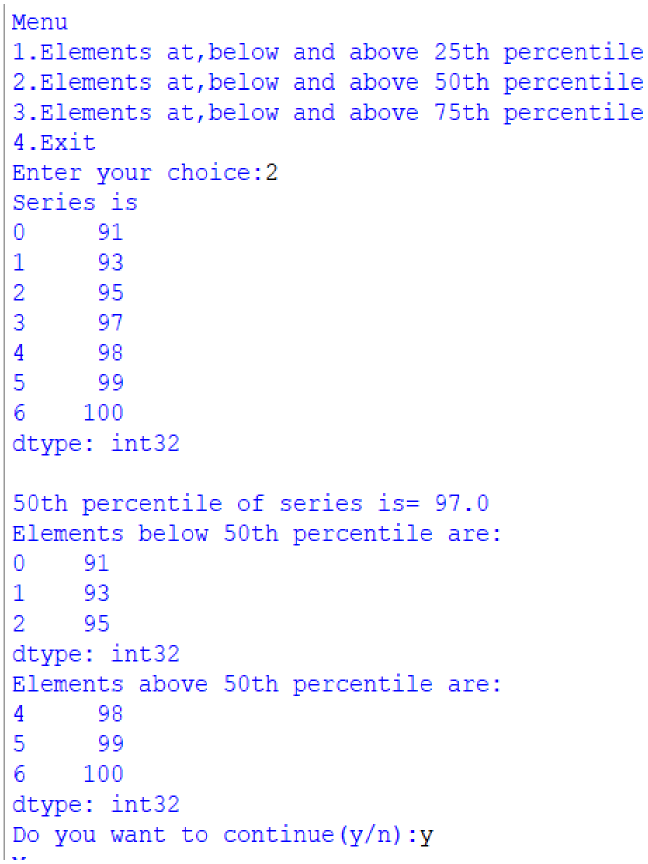


```
import pandas as pd
import numpy as np
s=pd.Series(np.array([91,93,95,97,98,99,100]))
opt='y'
while(opt=='y'):
    print("Menu")
    print("1.Elements at,below and above 25th percentile")
    print("2.Elements at,below and above 50th percentile")
    print("3.Elements at,below and above 75th percentile")
    print("4.Exit")
    ch=int(input("Enter your choice:"))
    if ch==1:
        print('Series is')
        print(s)
        res1=s.quantile(q=0.25)
        print('25th percentile of series is=',res1)
        print('Elements below 25th percentile are:')
        print(s[s<res1])
        print('Elements above 25th percentile are:')
        print(s[s>res1])
    if ch==2:
        print('Series is')
        print(s)
        res2=s.quantile(q=0.50)
        print('\n50th percentile of series is=',res2)
        print('Elements below 50th percentile are:')
        print(s[s<res2])
        print('Elements above 50th percentile are:')
        print(s[s>res2])
        -
    if ch==3:
        print('Series is')
        print(s)
        res3=s.quantile(q=0.75)
        print('\n75th percentile of series is=',res3)
        print('Elements below 75th percentile are:')
        print(s[s<res3])
        print('Elements above 75th percentile are:')
        print(s[s>res3])
    if ch==4:
        break
    opt=input('Do you want to continue(y/n):')
```

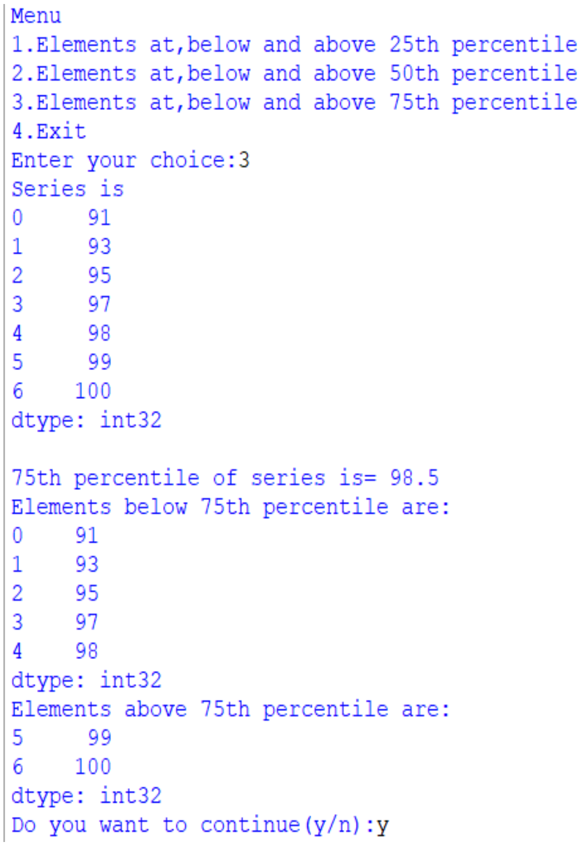
Output:



```
Menu
1.Elements at,below and above 25th percentile
2.Elements at,below and above 50th percentile
3.Elements at,below and above 75th percentile
4.Exit
Enter your choice:1
Series is
0      91
1      93
2      95
3      97
4      98
5      99
6     100
dtype: int32
25th percentile of series is= 94.0
Elements below 25th percentile are:
0     91
1     93
dtype: int32
Elements above 25th percentile are:
2      95
3      97
4      98
5      99
6     100
dtype: int32
Do you want to continue(y/n):y
```



```
Menu
1.Elements at,below and above 25th percentile
2.Elements at,below and above 50th percentile
3.Elements at,below and above 75th percentile
4.Exit
Enter your choice:2
Series is
0      91
1      93
2      95
3      97
4      98
5      99
6     100
dtype: int32

50th percentile of series is= 97.0
Elements below 50th percentile are:
0     91
1     93
2     95
dtype: int32
Elements above 50th percentile are:
4      98
5      99
6     100
dtype: int32
Do you want to continue(y/n):y
```



```
Menu
1.Elements at,below and above 25th percentile
2.Elements at,below and above 50th percentile
3.Elements at,below and above 75th percentile
4.Exit
Enter your choice:3
Series is
0      91
1      93
2      95
3      97
4      98
5      99
6     100
dtype: int32

75th percentile of series is= 98.5
Elements below 75th percentile are:
0     91
1     93
2     95
3     97
4     98
dtype: int32
Elements above 75th percentile are:
5      99
6     100
dtype: int32
Do you want to continue(y/n):y
```

Practical 5:

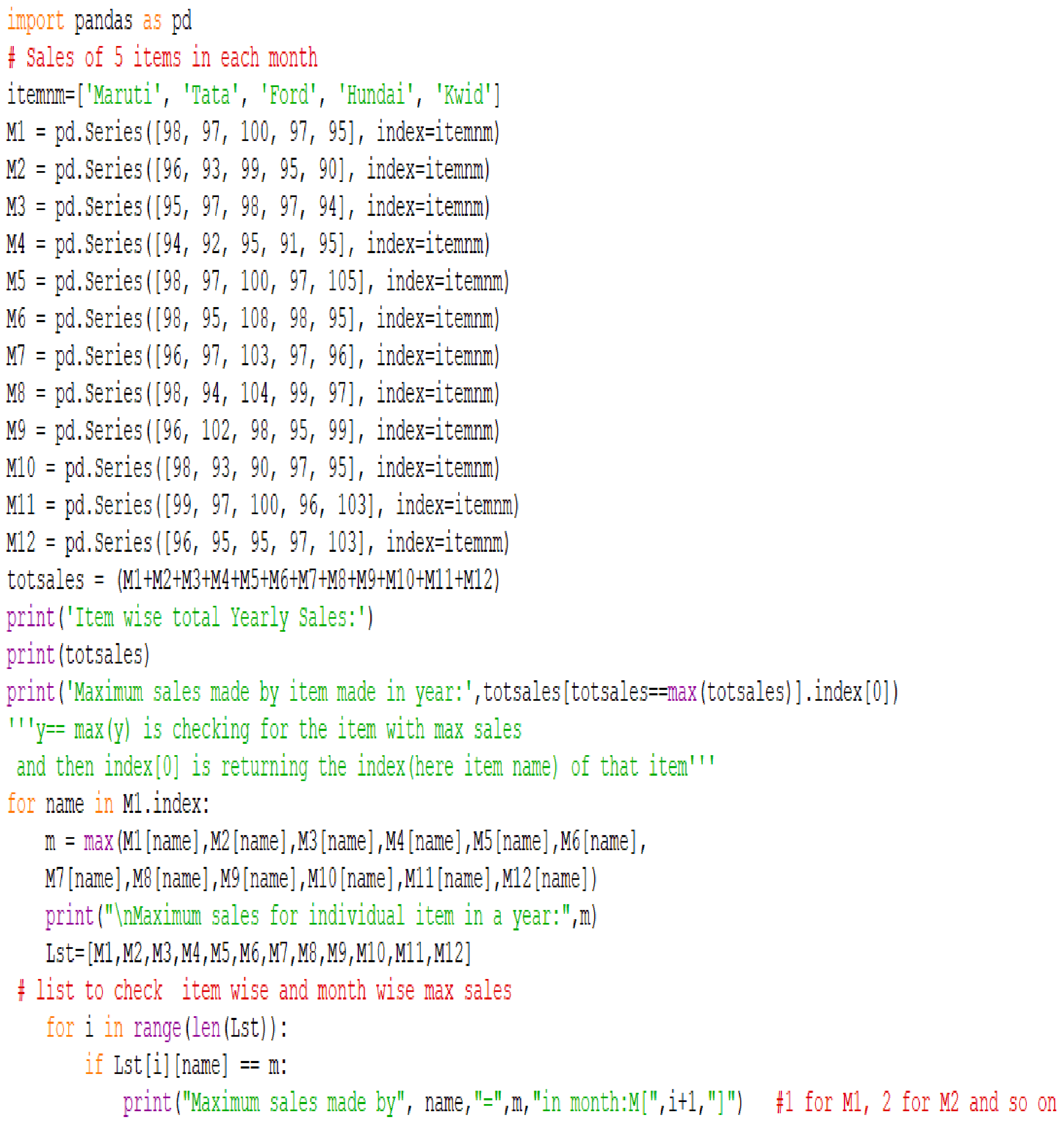
Write a program that stores the sales of 5 cars for each month in 12 Series objects. The program should display:

1) Item wise total yearly Sales

2) Maximum Sales of item made in a year

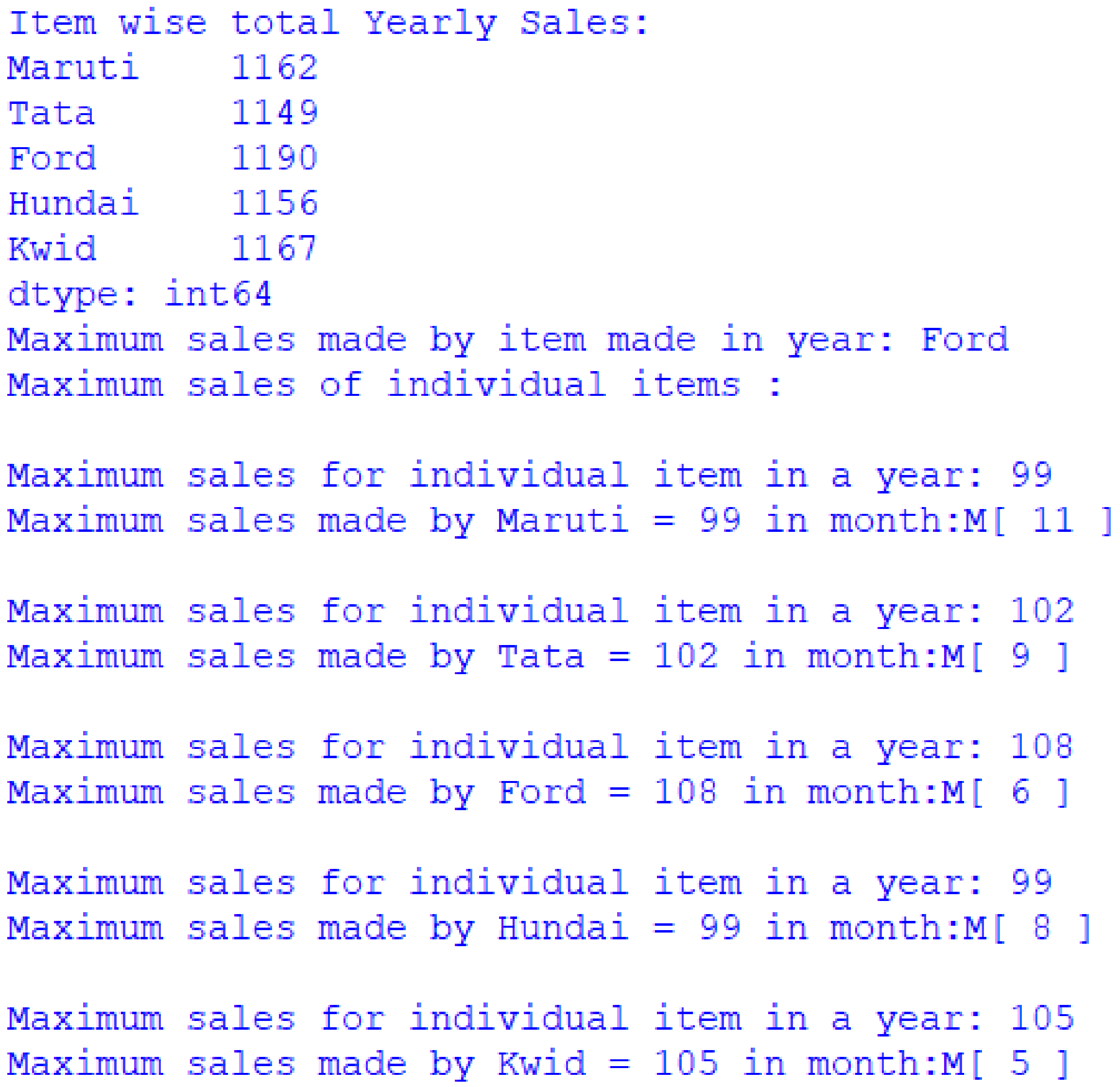
3) Maximum Sales for individual items

4) Maximum Sales of item made in a months.



```
import pandas as pd
# Sales of 5 items in each month
itemnm=['Maruti', 'Tata', 'Ford', 'Hundai', 'Kwid']
M1 = pd.Series([98, 97, 100, 97, 95], index=itemnm)
M2 = pd.Series([96, 93, 99, 95, 90], index=itemnm)
M3 = pd.Series([95, 97, 98, 97, 94], index=itemnm)
M4 = pd.Series([94, 92, 95, 91, 95], index=itemnm)
M5 = pd.Series([98, 97, 100, 97, 105], index=itemnm)
M6 = pd.Series([98, 95, 108, 98, 95], index=itemnm)
M7 = pd.Series([96, 97, 103, 97, 96], index=itemnm)
M8 = pd.Series([98, 94, 104, 99, 97], index=itemnm)
M9 = pd.Series([96, 102, 98, 95, 99], index=itemnm)
M10 = pd.Series([98, 93, 90, 97, 95], index=itemnm)
M11 = pd.Series([99, 97, 100, 96, 103], index=itemnm)
M12 = pd.Series([96, 95, 95, 97, 103], index=itemnm)
totsales = (M1+M2+M3+M4+M5+M6+M7+M8+M9+M10+M11+M12)
print('Item wise total Yearly Sales:')
print(totsales)
print('Maximum sales made by item made in year:',totsales[totsales==max(totsales)].index[0])
'''y== max(y) is checking for the item with max sales
 and then index[0] is returning the index(here item name) of that item'''
for name in M1.index:
    m = max(M1[name],M2[name],M3[name],M4[name],M5[name],M6[name],
    M7[name],M8[name],M9[name],M10[name],M11[name],M12[name])
    print("\nMaximum sales for individual item in a year:",m)
    Lst=[M1,M2,M3,M4,M5,M6,M7,M8,M9,M10,M11,M12]
 # list to check  item wise and month wise max sales
    for i in range(len(Lst)):
        if Lst[i][name] == m:
            print("Maximum sales made by", name,"=",m,"in month:M[",i+1,"]")   #1 for M1, 2 for M2 and so on
```

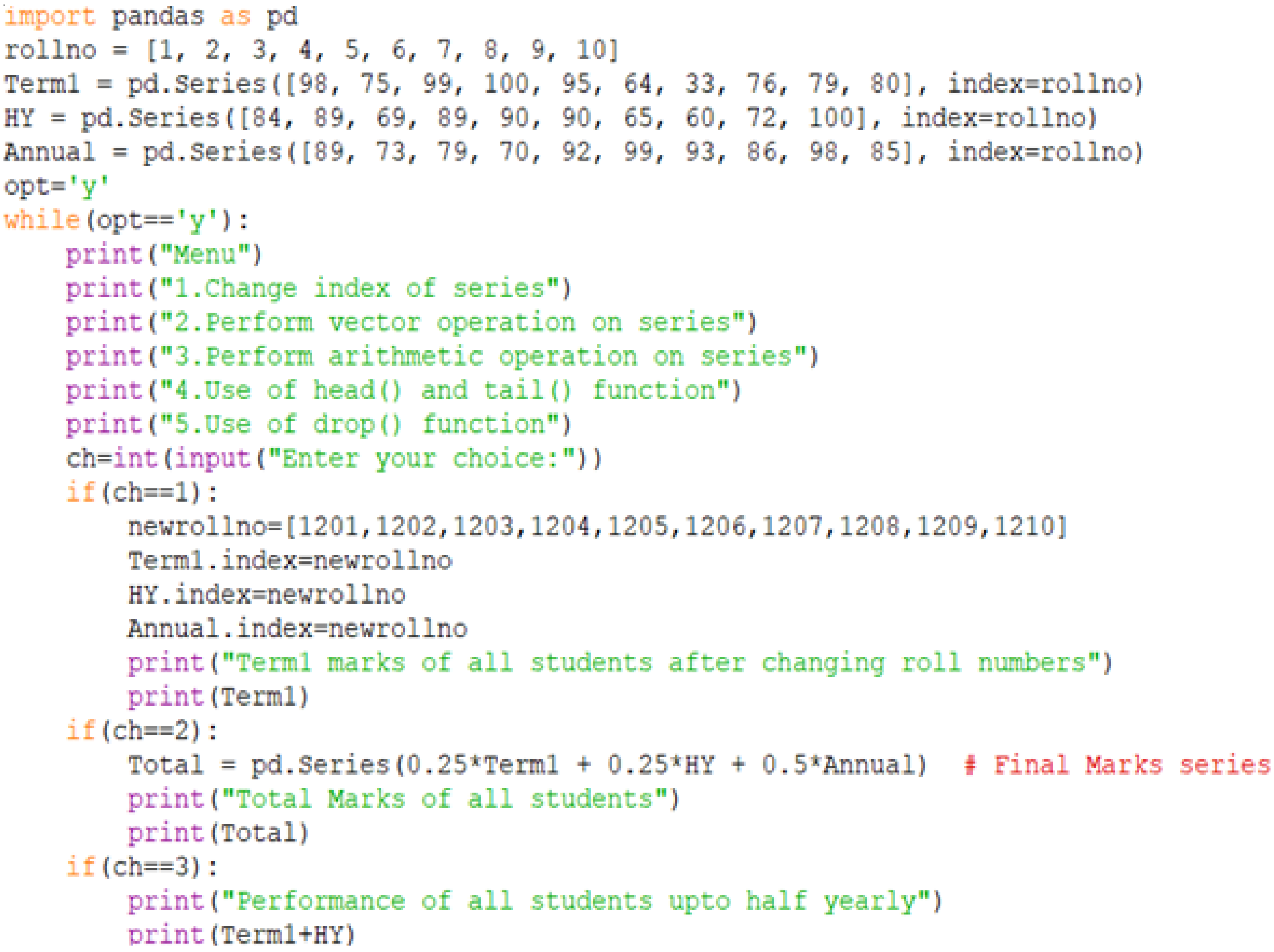
Output:



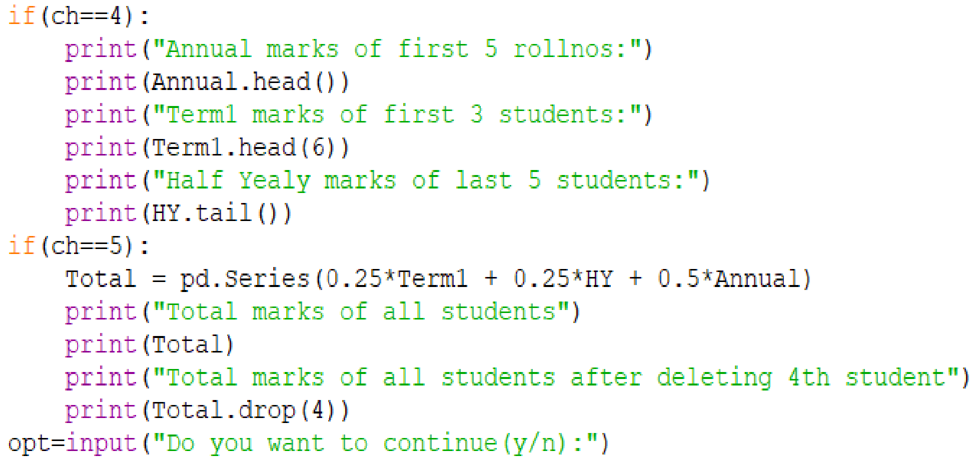
```
Item wise total Yearly Sales:
Maruti    1162
Tata      1149
Ford      1190
Hundai    1156
Kwid      1167
dtype: int64
Maximum sales made by item made in year: Ford
Maximum sales of individual items :

Maximum sales for individual item in a year: 99
Maximum sales made by Maruti = 99 in month:M[ 11 ]

Maximum sales for individual item in a year: 102
Maximum sales made by Tata = 102 in month:M[ 9 ]

Maximum sales for individual item in a year: 108
Maximum sales made by Ford = 108 in month:M[ 6 ]

Maximum sales for individual item in a year: 99
Maximum sales made by Hundai = 99 in month:M[ 8 ]

Maximum sales for individual item in a year: 105
Maximum sales made by Kwid = 105 in month:M[ 5 ]
```

Practical 6-Write a program to change index of existing series, perform vector, arithmetic operation on series and use of head (), tail (), drop () function

Solution:

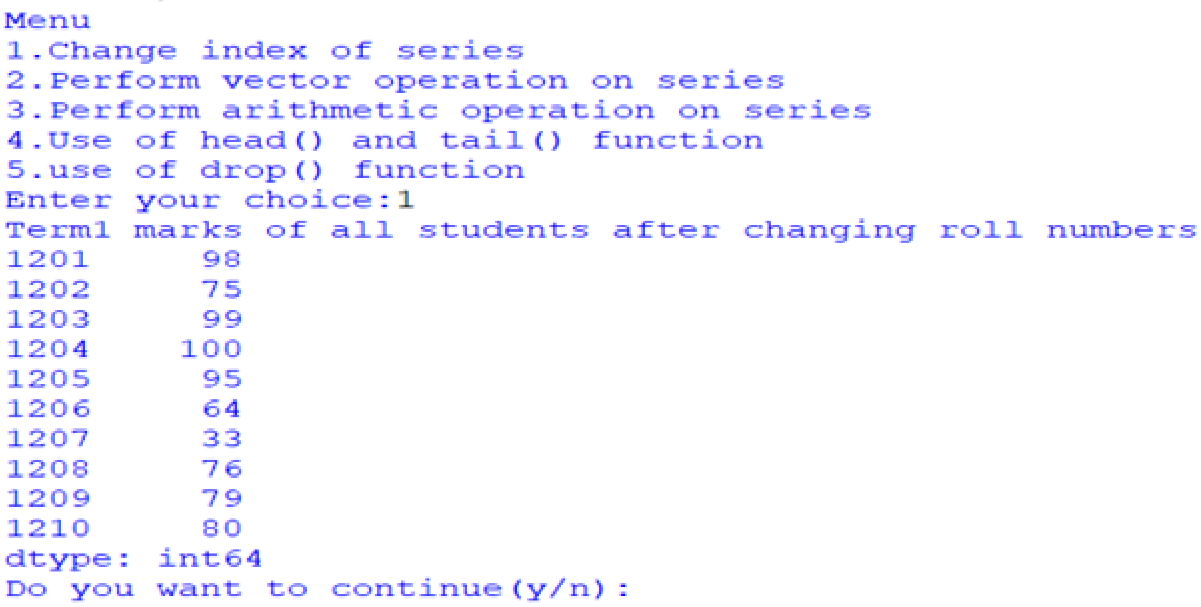


```
import pandas as pd
rollno = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
Term1 = pd.Series([98, 75, 99, 100, 95, 64, 33, 76, 79, 80], index=rollno)
HY = pd.Series([84, 89, 69, 89, 90, 90, 65, 60, 72, 100], index=rollno)
Annual = pd.Series([89, 73, 79, 70, 92, 99, 93, 86, 98, 85], index=rollno)
opt='y'
while(opt=='y'):
    print("Menu")
    print("1.Change index of series")
    print("2.Perform vector operation on series")
    print("3.Perform arithmetic operation on series")
    print("4.Use of head() and tail() function")
    print("5.Use of drop() function")
    ch=int(input("Enter your choice:"))
    if(ch==1):
        newrollno=[1201,1202,1203,1204,1205,1206,1207,1208,1209,1210]
        Term1.index=newrollno
        HY.index=newrollno
        Annual.index=newrollno
        print("Term1 marks of all students after changing roll numbers")
        print(Term1)
    if(ch==2):
        Total = pd.Series(0.25*Term1 + 0.25*HY + 0.5*Annual)  # Final Marks series
        print("Total Marks of all students")
        print(Total)
    if(ch==3):
        print("Performance of all students upto half yearly")
        print(Term1+HY)
```

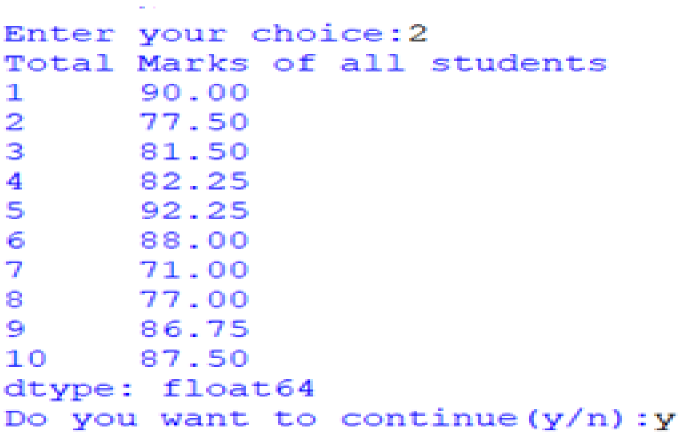


```
    if(ch==4):
        print("Annual marks of first 5 rollnos:")
        print(Annual.head())
        print("Term1 marks of first 3 students:")
        print(Term1.head(6))
        print("Half Yealy marks of last 5 students:")
        print(HY.tail())
    if(ch==5):
        Total = pd.Series(0.25*Term1 + 0.25*HY + 0.5*Annual)
        print("Total marks of all students")
        print(Total)
        print("Total marks of all students after deleting 4th student")
        print(Total.drop(4))
    opt=input("Do you want to continue(y/n):")
```

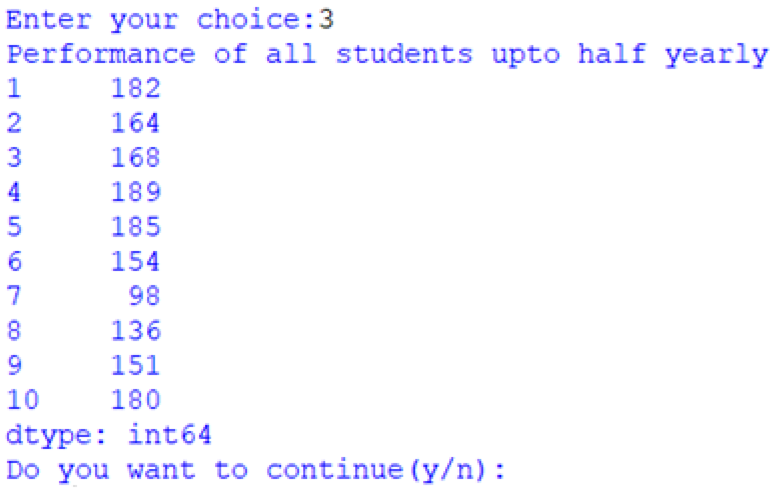
Output:



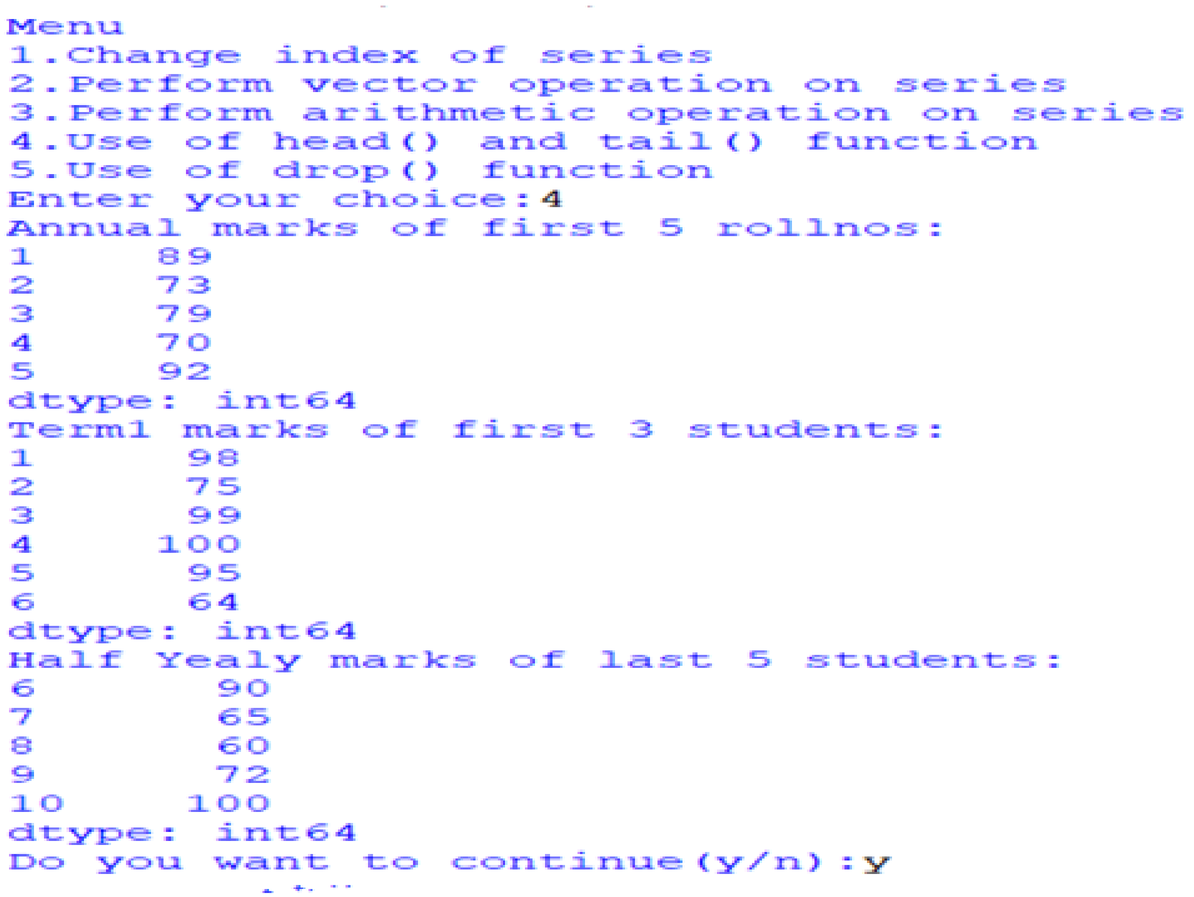
```
Menu
1.Change index of series
2.Perform vector operation on series
3.Perform arithmetic operation on series
4.Use of head() and tail() function
5.use of drop() function
Enter your choice:1
Term1 marks of all students after changing roll numbers
1201      98
1202      75
1203      99
1204     100
1205      95
1206      64
1207      33
1208      76
1209      79
1210      80
dtype: int64
Do you want to continue(y/n):
```



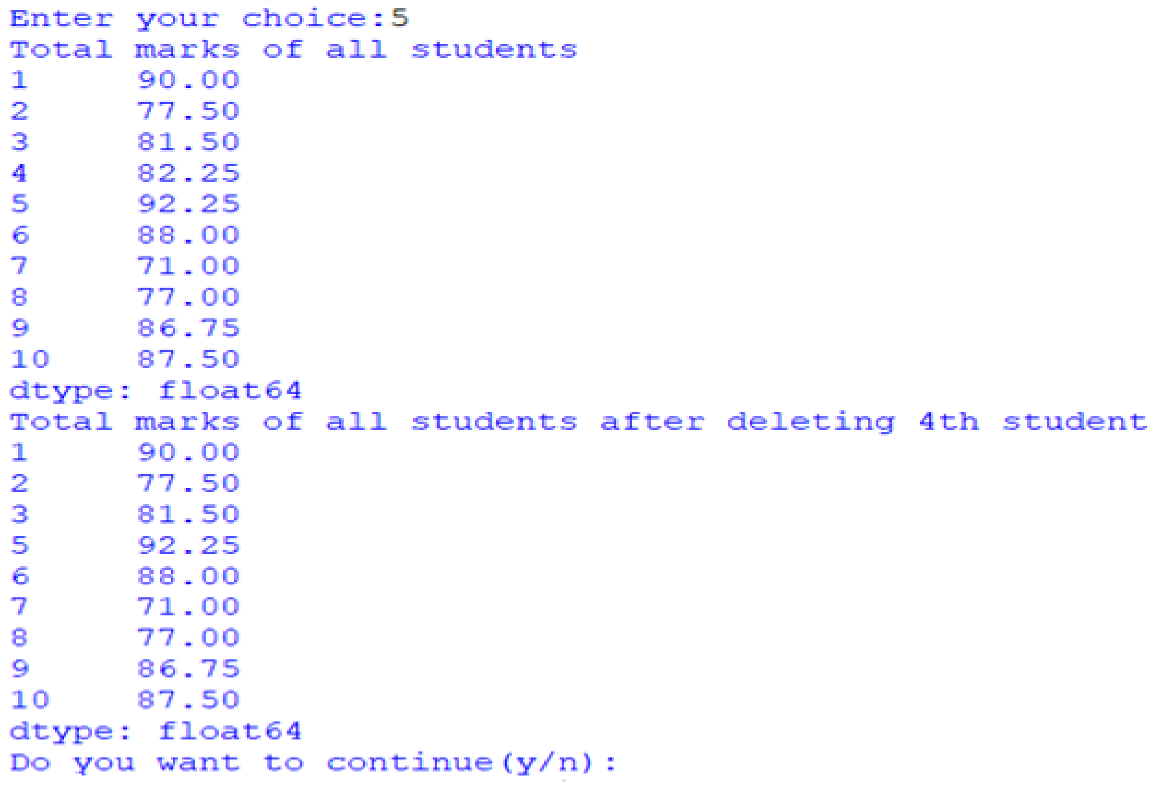
```
Enter your choice:2
Total Marks of all students
1      90.00
2      77.50
3      81.50
4      82.25
5      92.25
6      88.00
7      71.00
8      77.00
9      86.75
10     87.50
dtype: float64
Do you want to continue(y/n):y
```



```
Enter your choice:3
Performance of all students upto half yearly
1     182
2     164
3     168
4     189
5     185
6     154
7      98
8     136
9     151
10    180
dtype: int64
Do you want to continue(y/n):
```



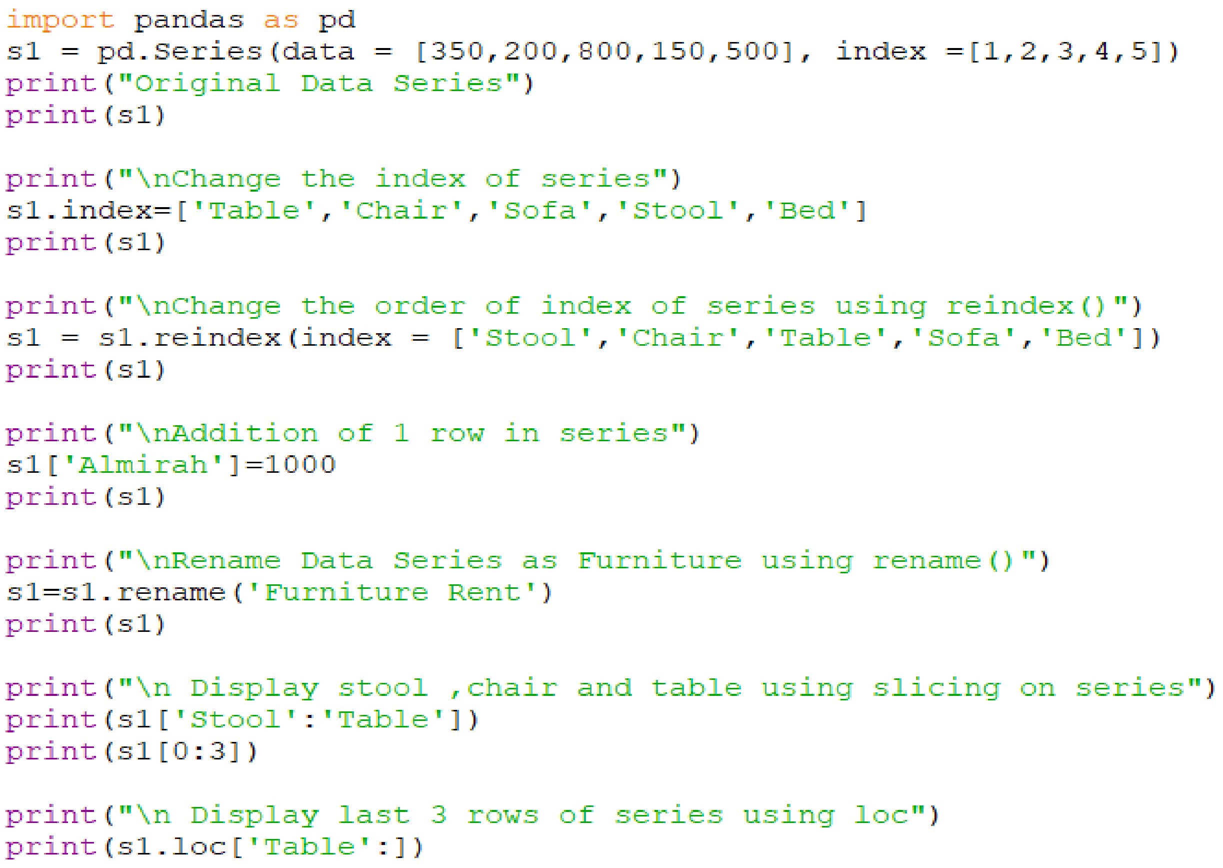
```
Menu
1.Change index of series
2.Perform vector operation on series
3.Perform arithmetic operation on series
4.Use of head() and tail() function
5.Use of drop() function
Enter your choice:4
Annual marks of first 5 rollnos:
1     89
2     73
3     79
4     70
5     92
dtype: int64
Term1 marks of first 3 students:
1      98
2      75
3      99
4     100
5      95
6      64
dtype: int64
Half Yealy marks of last 5 students:
6       90
7       65
8       60
9       72
10     100
dtype: int64
Do you want to continue(y/n):y
```



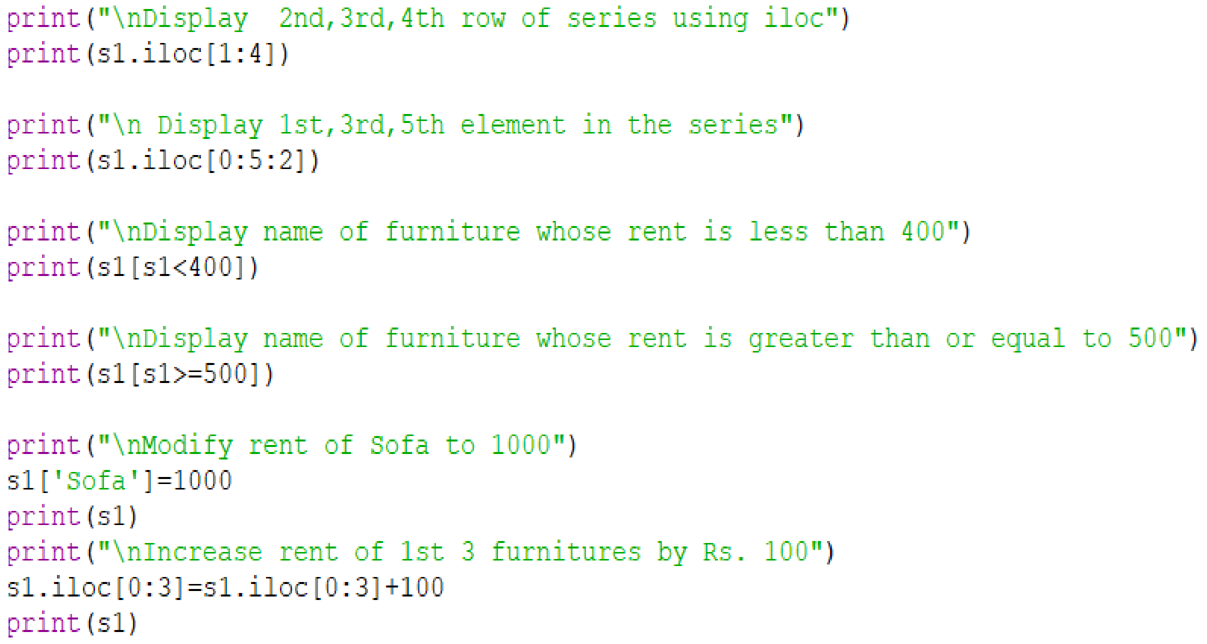
```
Enter your choice:5
Total marks of all students
1     90.00
2     77.50
3     81.50
4     82.25
5     92.25
6     88.00
7     71.00
8     77.00
9     86.75
10    87.50
dtype: float64
Total marks of all students after deleting 4th student
1     90.00
2     77.50
3     81.50
5     92.25
6     88.00
7     71.00
8     77.00
9     86.75
10    87.50
dtype: float64
Do you want to continue(y/n):
```

Practical 7- Write a program to perform selection, indexing and slicing operation on Series.

Solution:

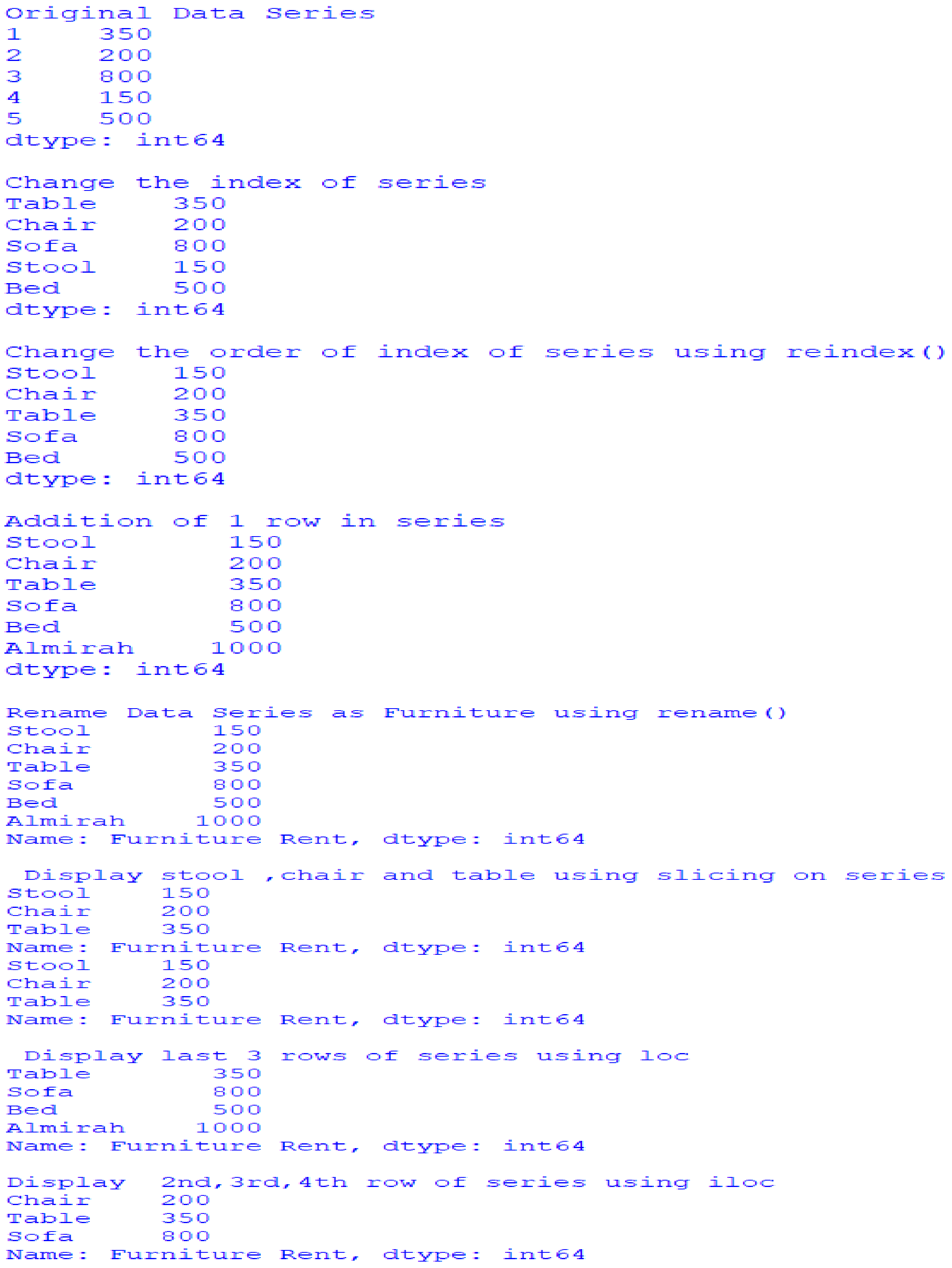


```
import pandas as pd
s1 = pd.Series(data = [350,200,800,150,500], index =[1,2,3,4,5])
print("Original Data Series")
print(s1)

print("\nChange the index of series")
s1.index=['Table','Chair','Sofa','Stool','Bed']
print(s1)

print("\nChange the order of index of series using reindex()")
s1 = s1.reindex(index = ['Stool','Chair','Table','Sofa','Bed'])
print(s1)

print("\nAddition of 1 row in series")
s1['Almirah']=1000
print(s1)

print("\nRename Data Series as Furniture using rename()")
s1=s1.rename('Furniture Rent')
print(s1)

print("\n Display stool ,chair and table using slicing on series")
print(s1['Stool':'Table'])
print(s1[0:3])

print("\n Display last 3 rows of series using loc")
print(s1.loc['Table':])
```

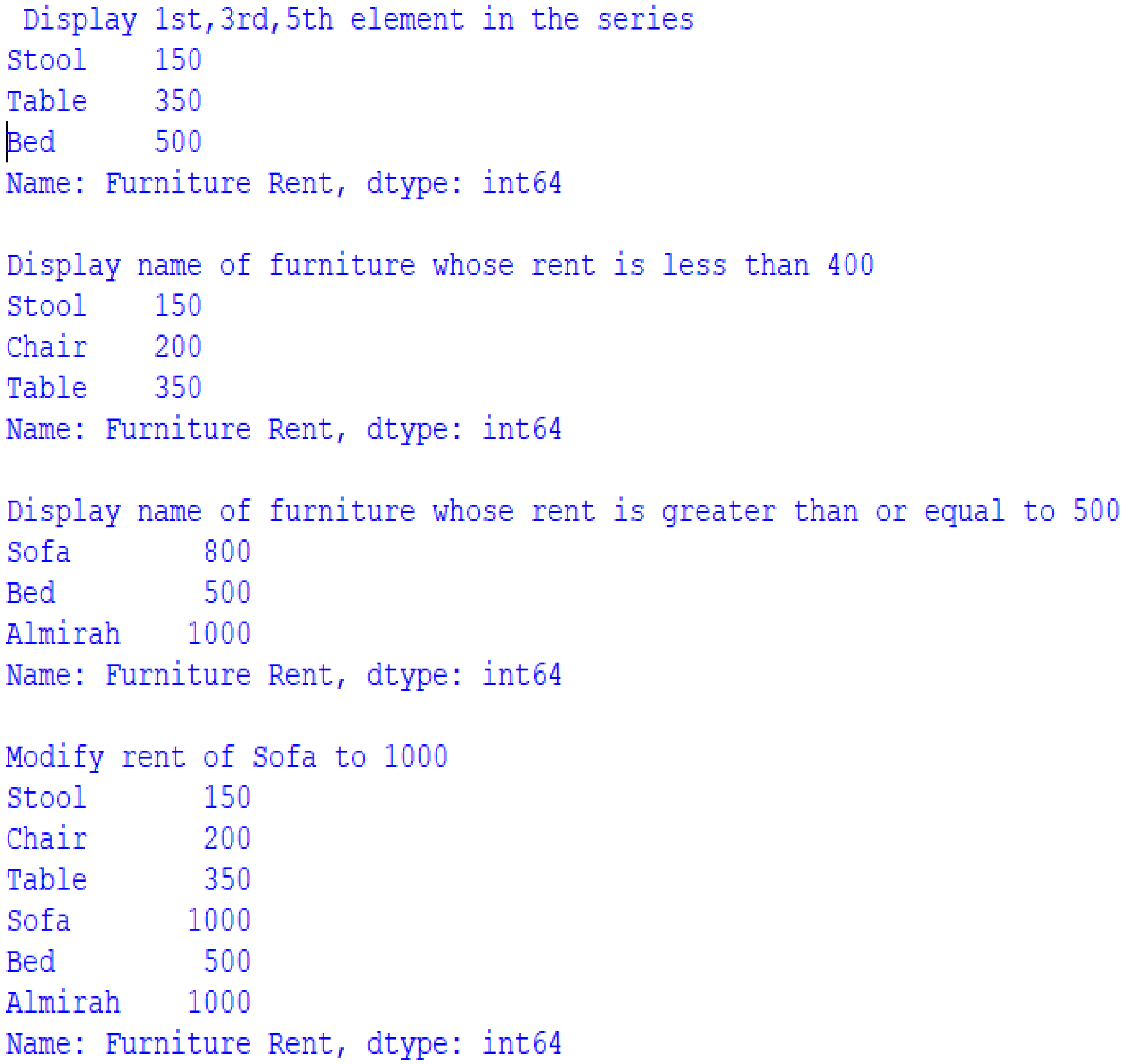


```
print("\nDisplay  2nd,3rd,4th row of series using iloc")
print(s1.iloc[1:4])

print("\n Display 1st,3rd,5th element in the series")
print(s1.iloc[0:5:2])

print("\nDisplay name of furniture whose rent is less than 400")
print(s1[s1<400])

print("\nDisplay name of furniture whose rent is greater than or equal to 500")
print(s1[s1>=500])

print("\nModify rent of Sofa to 1000")
s1['Sofa']=1000
print(s1)
print("\nIncrease rent of 1st 3 furnitures by Rs. 100")
s1.iloc[0:3]=s1.iloc[0:3]+100
print(s1)
```

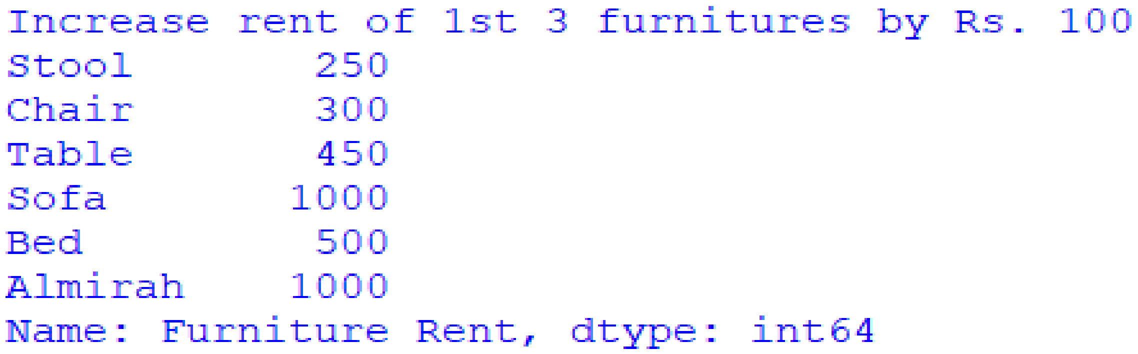
Output:



```
Original Data Series
1     350
2     200
3     800
4     150
5     500
dtype: int64

Change the index of series
Table     350
Chair     200
Sofa      800
Stool     150
Bed       500
dtype: int64

Change the order of index of series using reindex()
Stool     150
Chair     200
Table     350
Sofa      800
Bed       500
dtype: int64

Addition of 1 row in series
Stool        150
Chair        200
Table        350
Sofa         800
Bed          500
Almirah     1000
dtype: int64

Rename Data Series as Furniture using rename()
Stool        150
Chair        200
Table        350
Sofa         800
Bed          500
Almirah     1000
Name: Furniture Rent, dtype: int64

 Display stool ,chair and table using slicing on series
Stool     150
Chair     200
Table     350
Name: Furniture Rent, dtype: int64
Stool     150
Chair     200
Table     350
Name: Furniture Rent, dtype: int64

 Display last 3 rows of series using loc
Table        350
Sofa         800
Bed          500
Almirah     1000
Name: Furniture Rent, dtype: int64

Display  2nd,3rd,4th row of series using iloc
Chair     200
Table     350
Sofa      800
Name: Furniture Rent, dtype: int64
```



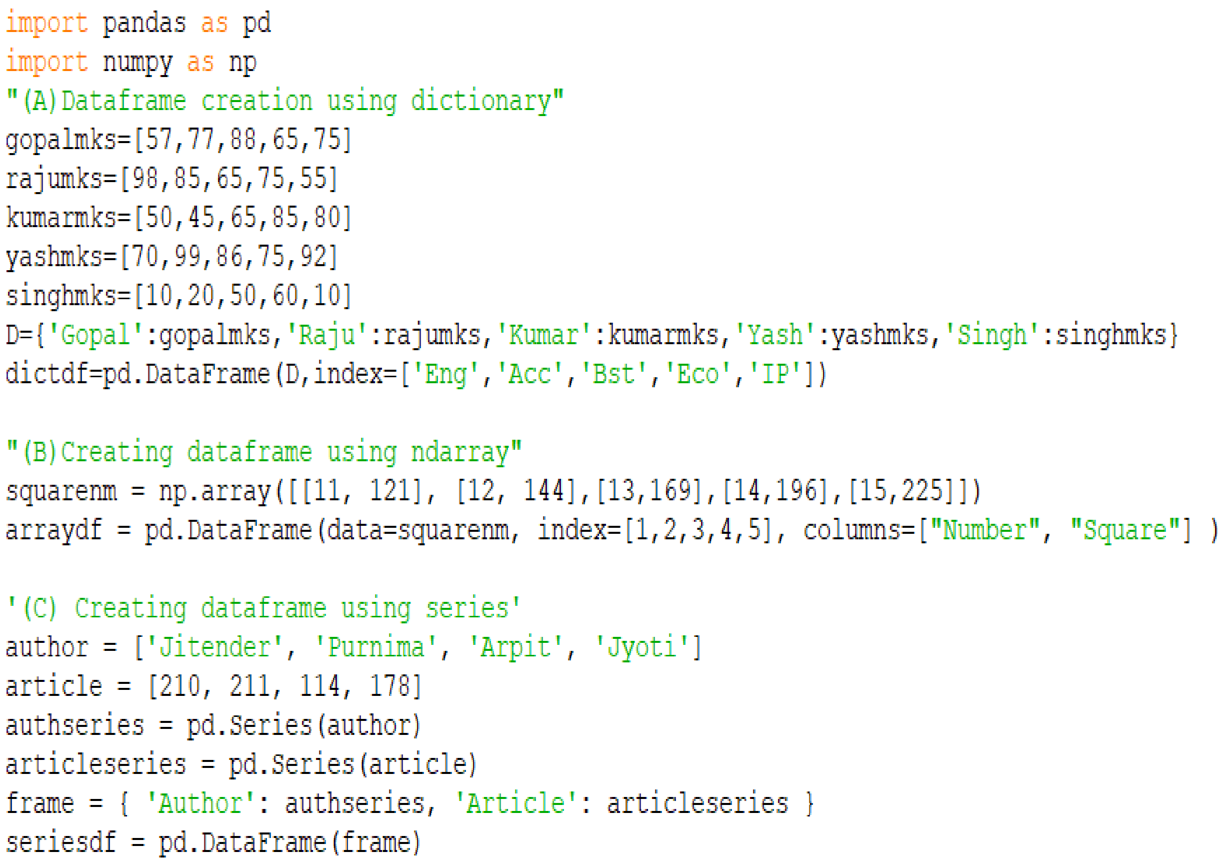
```
 Display 1st,3rd,5th element in the series
Stool    150
Table    350
Bed      500
Name: Furniture Rent, dtype: int64

Display name of furniture whose rent is less than 400
Stool    150
Chair    200
Table    350
Name: Furniture Rent, dtype: int64

Display name of furniture whose rent is greater than or equal to 500
Sofa        800
Bed         500
Almirah    1000
Name: Furniture Rent, dtype: int64

Modify rent of Sofa to 1000
Stool       150
Chair       200
Table       350
Sofa       1000
Bed         500
Almirah    1000
Name: Furniture Rent, dtype: int64
```



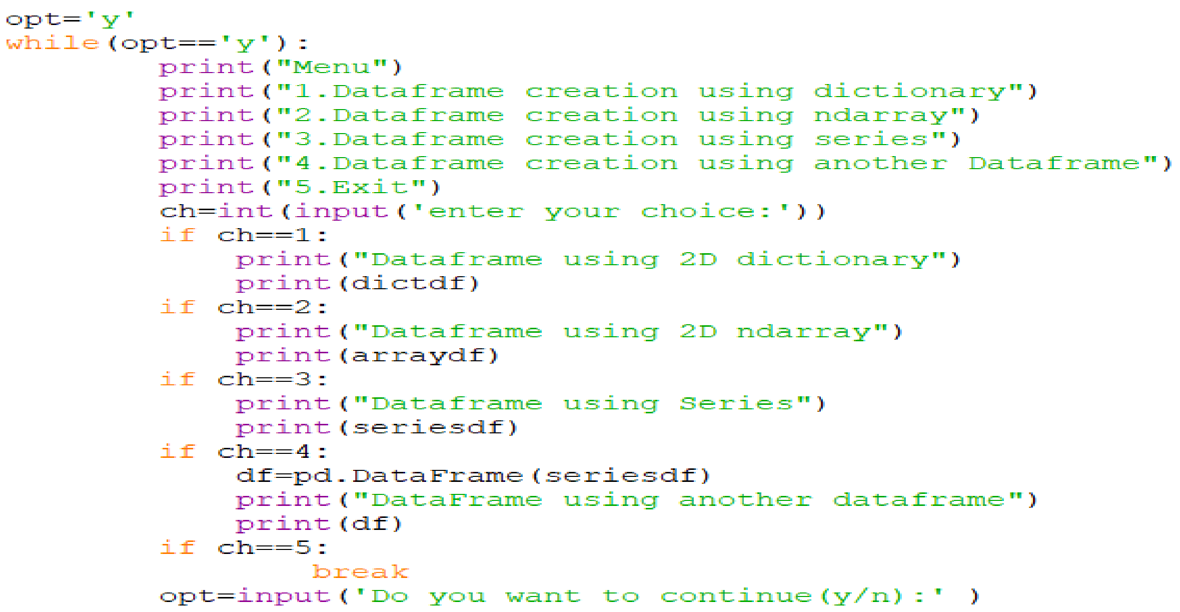
```
Increase rent of 1st 3 furnitures by Rs. 100
Stool       250
Chair       300
Table       450
Sofa       1000
Bed         500
Almirah    1000
Name: Furniture Rent, dtype: int64
```

Practical 8- A menu driven program to create a dataframe using 2D dictionary, 2D array, Series and another dataframe.

Solution:

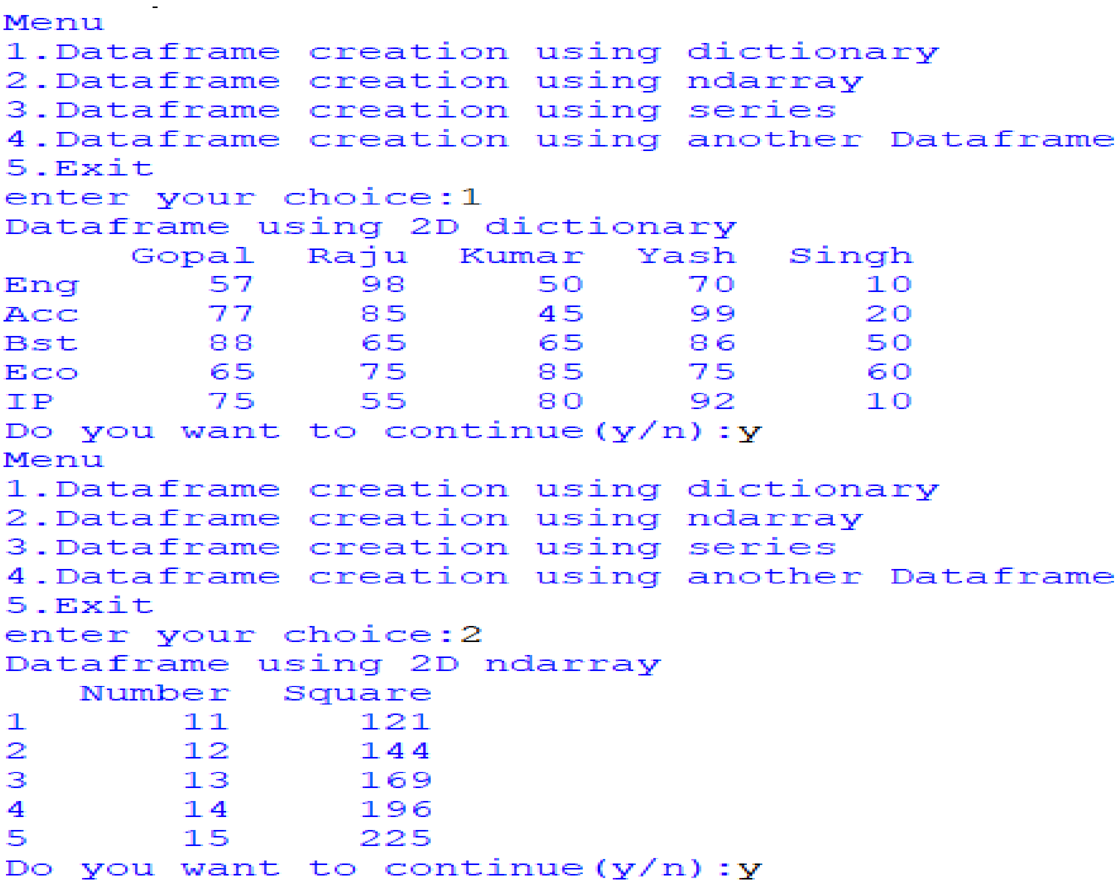


```python
import pandas as pd
import numpy as np
"(A)Dataframe creation using dictionary"
gopalmks=[57,77,88,65,75]
rajumks=[98,85,65,75,55]
kumarmks=[50,45,65,85,80]
yashmks=[70,99,86,75,92]
singhmks=[10,20,50,60,10]
D={'Gopal':gopalmks,'Raju':rajumks,'Kumar':kumarmks,'Yash':yashmks,'Singh':singhmks}
dictdf=pd.DataFrame(D,index=['Eng','Acc','Bst','Eco','IP'])

"(B)Creating dataframe using ndarray"
squarenm = np.array([[11, 121], [12, 144],[13,169],[14,196],[15,225]])
arraydf = pd.DataFrame(data=squarenm, index=[1,2,3,4,5], columns=["Number", "Square"] )

'(C) Creating dataframe using series'
author = ['Jitender', 'Purnima', 'Arpit', 'Jyoti']
article = [210, 211, 114, 178]
authseries = pd.Series(author)
articleseries = pd.Series(article)
frame = { 'Author': authseries, 'Article': articleseries }
seriesdf = pd.DataFrame(frame)
```

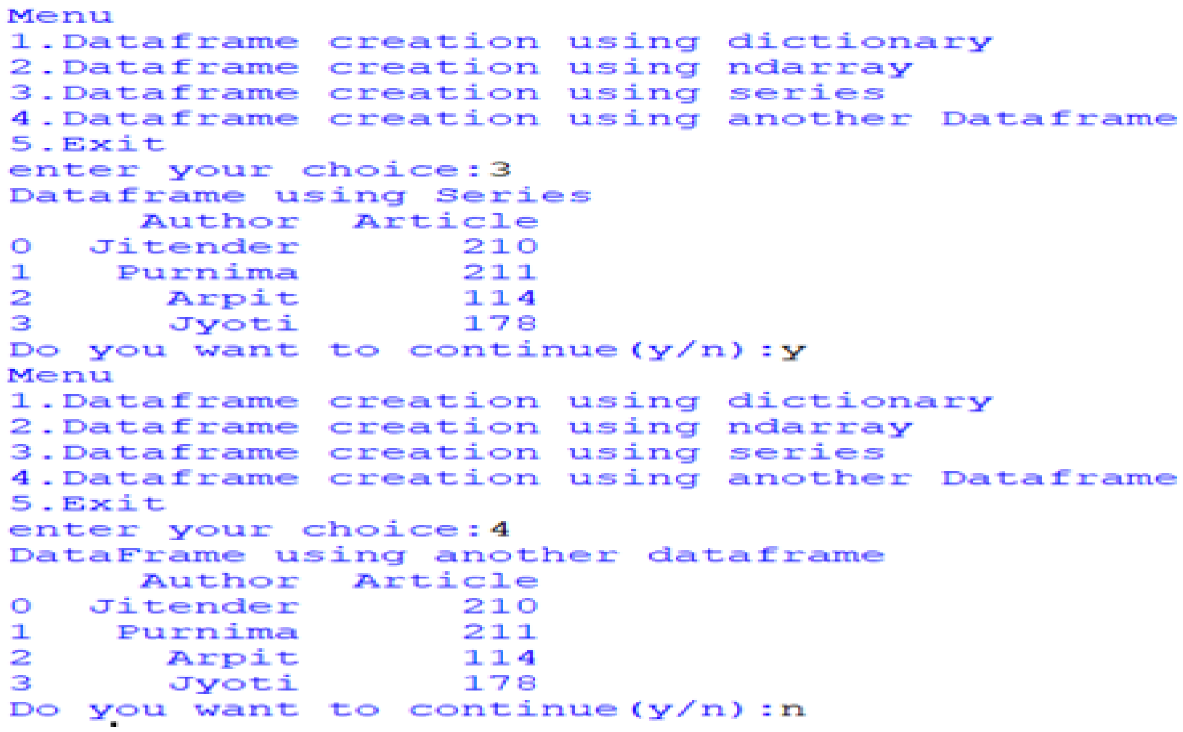


```python
opt='y'
while(opt=='y'):
        print("Menu")
        print("1.Dataframe creation using dictionary")
        print("2.Dataframe creation using ndarray")
        print("3.Dataframe creation using series")
        print("4.Dataframe creation using another Dataframe")
        print("5.Exit")
        ch=int(input('enter your choice:'))
        if ch==1:
            print("Dataframe using 2D dictionary")
            print(dictdf)
        if ch==2:
            print("Dataframe using 2D ndarray")
            print(arraydf)
        if ch==3:
            print("Dataframe using Series")
            print(seriesdf)
        if ch==4:
            df=pd.DataFrame(seriesdf)
            print("DataFrame using another dataframe")
            print(df)
        if ch==5:
                break
        opt=input('Do you want to continue(y/n):' )
```

Output:



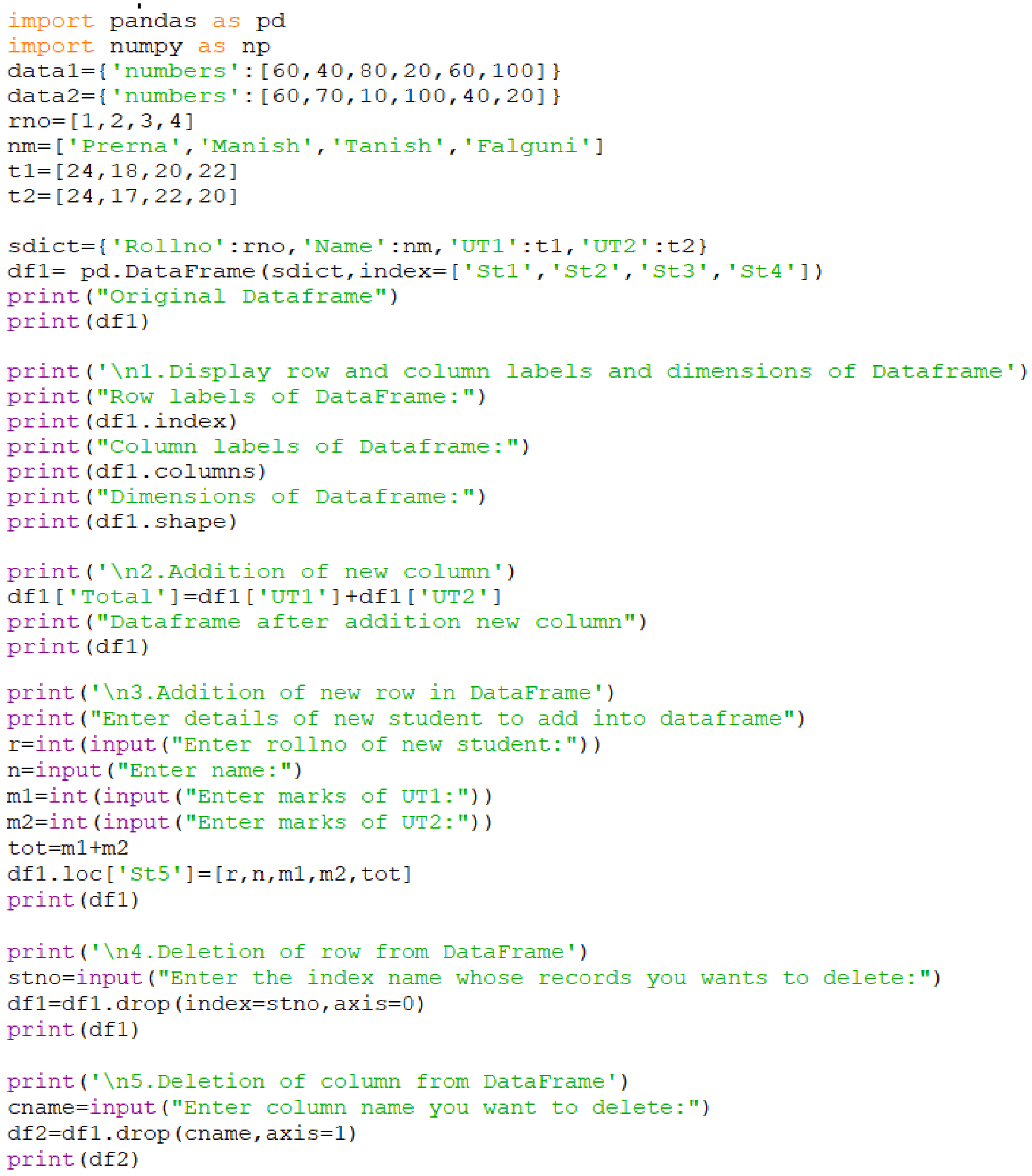
```
Menu
1.Dataframe creation using dictionary
2.Dataframe creation using ndarray
3.Dataframe creation using series
4.Dataframe creation using another Dataframe
5.Exit
enter your choice:1
Dataframe using 2D dictionary
     Gopal  Raju  Kumar  Yash  Singh
Eng     57    98     50    70     10
Acc     77    85     45    99     20
Bst     88    65     65    86     50
Eco     65    75     85    75     60
IP      75    55     80    92     10
Do you want to continue(y/n):y
Menu
1.Dataframe creation using dictionary
2.Dataframe creation using ndarray
3.Dataframe creation using series
4.Dataframe creation using another Dataframe
5.Exit
enter your choice:2
Dataframe using 2D ndarray
   Number  Square
1      11     121
2      12     144
3      13     169
4      14     196
5      15     225
Do you want to continue(y/n):y
```



```
Menu
1.Dataframe creation using dictionary
2.Dataframe creation using ndarray
3.Dataframe creation using series
4.Dataframe creation using another Dataframe
5.Exit
enter your choice:3
Dataframe using Series
     Author  Article
0  Jitender      210
1   Purnima      211
2     Arpit      114
3     Jyoti      178
Do you want to continue(y/n):y
Menu
1.Dataframe creation using dictionary
2.Dataframe creation using ndarray
3.Dataframe creation using series
4.Dataframe creation using another Dataframe
5.Exit
enter your choice:4
DataFrame using another dataframe
     Author  Article
0  Jitender      210
1   Purnima      211
2     Arpit      114
3     Jyoti      178
Do you want to continue(y/n):n
```

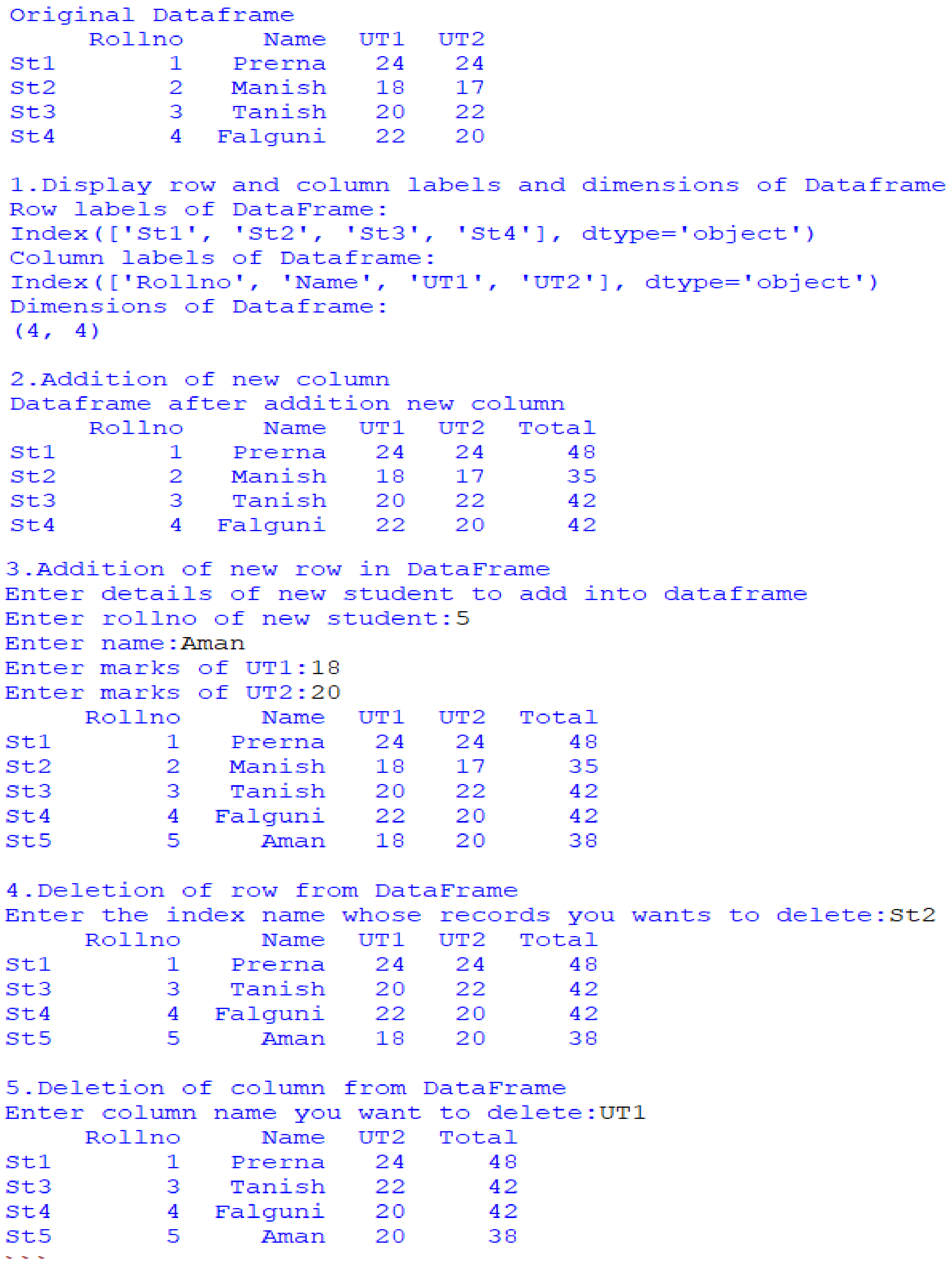
Practical 9- Create a data frame for examination result and 1. Display row labels, column labels, data types of each column and the dimensions. 2. Add and delete of column and row in Dataframe

Solution:



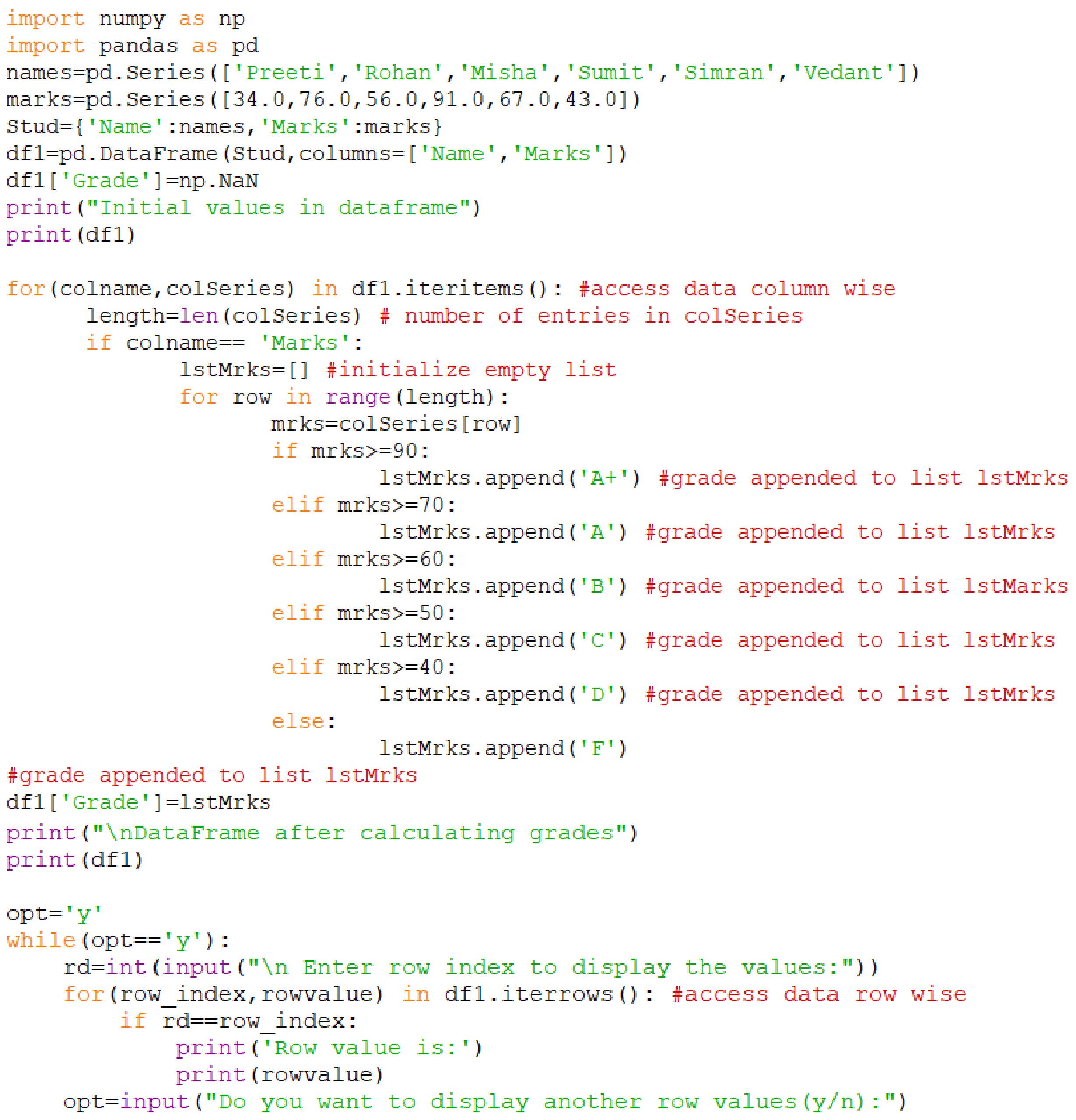
```
import pandas as pd
import numpy as np
data1={'numbers':[60,40,80,20,60,100]}
data2={'numbers':[60,70,10,100,40,20]}
rno=[1,2,3,4]
nm=['Prerna','Manish','Tanish','Falguni']
t1=[24,18,20,22]
t2=[24,17,22,20]

sdict={'Rollno':rno,'Name':nm,'UT1':t1,'UT2':t2}
df1= pd.DataFrame(sdict,index=['St1','St2','St3','St4'])
print("Original Dataframe")
print(df1)

print('\n1.Display row and column labels and dimensions of Dataframe')
print("Row labels of DataFrame:")
print(df1.index)
print("Column labels of Dataframe:")
print(df1.columns)
print("Dimensions of Dataframe:")
print(df1.shape)

print('\n2.Addition of new column')
df1['Total']=df1['UT1']+df1['UT2']
print("Dataframe after addition new column")
print(df1)

print('\n3.Addition of new row in DataFrame')
print("Enter details of new student to add into dataframe")
r=int(input("Enter rollno of new student:"))
n=input("Enter name:")
m1=int(input("Enter marks of UT1:"))
m2=int(input("Enter marks of UT2:"))
tot=m1+m2
df1.loc['St5']=[r,n,m1,m2,tot]
print(df1)

print('\n4.Deletion of row from DataFrame')
stno=input("Enter the index name whose records you wants to delete:")
df1=df1.drop(index=stno,axis=0)
print(df1)

print('\n5.Deletion of column from DataFrame')
cname=input("Enter column name you want to delete:")
df2=df1.drop(cname,axis=1)
print(df2)
```

Output:



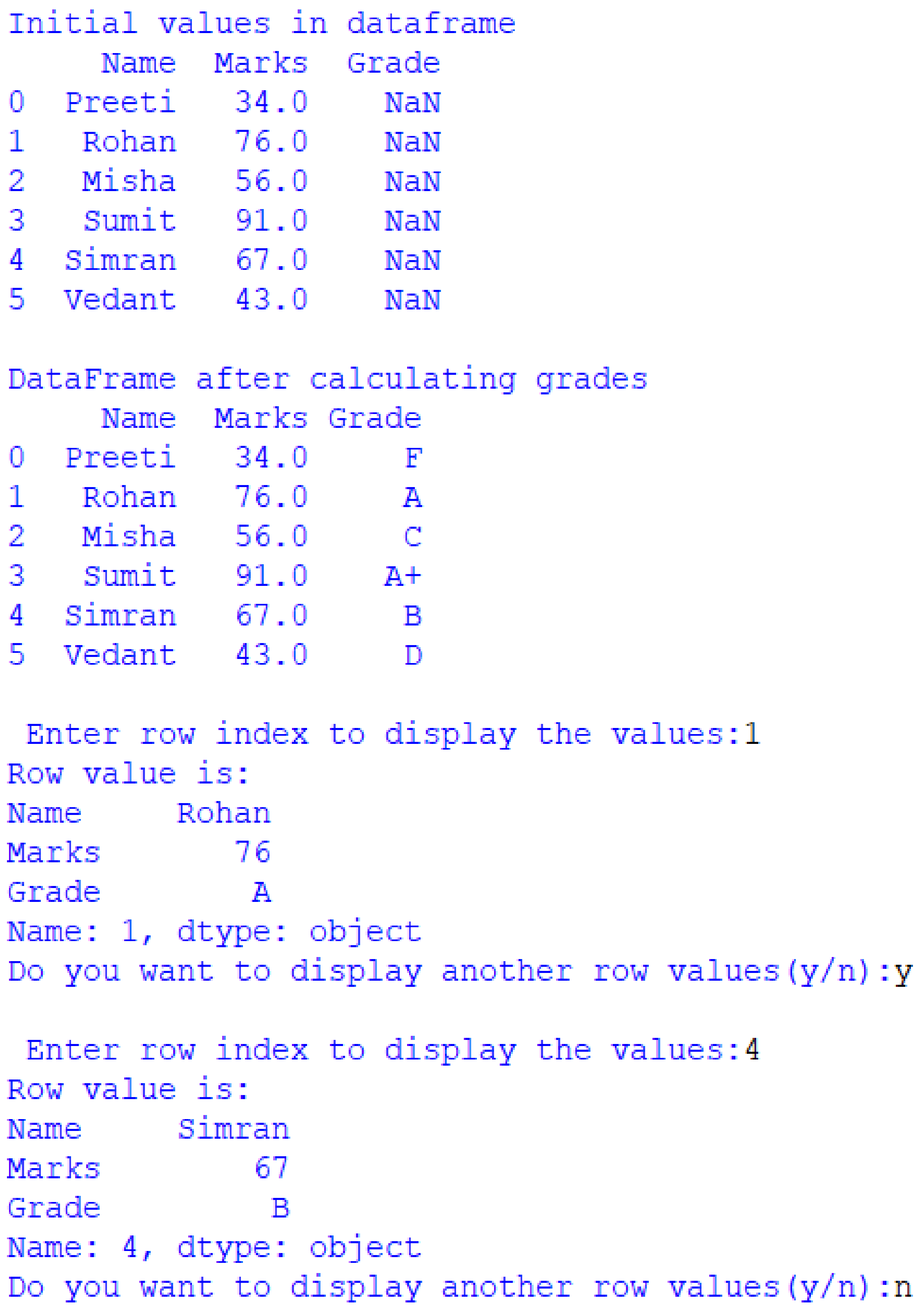
```
Original Dataframe
     Rollno     Name  UT1  UT2
St1       1   Prerna   24   24
St2       2   Manish   18   17
St3       3   Tanish   20   22
St4       4  Falguni   22   20

1.Display row and column labels and dimensions of Dataframe
Row labels of DataFrame:
Index(['St1', 'St2', 'St3', 'St4'], dtype='object')
Column labels of Dataframe:
Index(['Rollno', 'Name', 'UT1', 'UT2'], dtype='object')
Dimensions of Dataframe:
(4, 4)

2.Addition of new column
Dataframe after addition new column
     Rollno     Name  UT1  UT2  Total
St1       1   Prerna   24   24     48
St2       2   Manish   18   17     35
St3       3   Tanish   20   22     42
St4       4  Falguni   22   20     42

3.Addition of new row in DataFrame
Enter details of new student to add into dataframe
Enter rollno of new student:5
Enter name:Aman
Enter marks of UT1:18
Enter marks of UT2:20
     Rollno     Name  UT1  UT2  Total
St1       1   Prerna   24   24     48
St2       2   Manish   18   17     35
St3       3   Tanish   20   22     42
St4       4  Falguni   22   20     42
St5       5     Aman   18   20     38

4.Deletion of row from DataFrame
Enter the index name whose records you wants to delete:St2
     Rollno     Name  UT1  UT2  Total
St1       1   Prerna   24   24     48
St3       3   Tanish   20   22     42
St4       4  Falguni   22   20     42
St5       5     Aman   18   20     38

5.Deletion of column from DataFrame
Enter column name you want to delete:UT1
     Rollno     Name  UT2  Total
St1       1   Prerna   24     48
St3       3   Tanish   22     42
St4       4  Falguni   20     42
St5       5     Aman   20     38
```

Practical 10- Write a program to iterate over a DataFrame containing names and marks, then calculates grades as per marks (as per guideline below) and adds them to the grade column.
Marks > =90 Grade A+
Marks 70 – 90 Grade A
Marks 60 – 70 Grade B
Marks 50 – 60 Grade C
Marks 40 – 50 Grade D
Marks < 40 Grade F
Solution:



```
import numpy as np
import pandas as pd
names=pd.Series(['Preeti','Rohan','Misha','Sumit','Simran','Vedant'])
marks=pd.Series([34.0,76.0,56.0,91.0,67.0,43.0])
Stud={'Name':names,'Marks':marks}
df1=pd.DataFrame(Stud,columns=['Name','Marks'])
df1['Grade']=np.NaN
print("Initial values in dataframe")
print(df1)

for(colname,colSeries) in df1.iteritems(): #access data column wise
      length=len(colSeries) # number of entries in colSeries
      if colname== 'Marks':
             lstMrks=[] #initialize empty list
             for row in range(length):
                    mrks=colSeries[row]
                    if mrks>=90:
                            lstMrks.append('A+') #grade appended to list lstMrks
                    elif mrks>=70:
                            lstMrks.append('A') #grade appended to list lstMrks
                    elif mrks>=60:
                            lstMrks.append('B') #grade appended to list lstMarks
                    elif mrks>=50:
                            lstMrks.append('C') #grade appended to list lstMrks
                    elif mrks>=40:
                            lstMrks.append('D') #grade appended to list lstMrks
                    else:
                            lstMrks.append('F')
#grade appended to list lstMrks
df1['Grade']=lstMrks
print("\nDataFrame after calculating grades")
print(df1)

opt='y'
while(opt=='y'):
    rd=int(input("\n Enter row index to display the values:"))
    for(row_index,rowvalue) in df1.iterrows(): #access data row wise
        if rd==row_index:
            print('Row value is:')
            print(rowvalue)
    opt=input("Do you want to display another row values(y/n):")
```

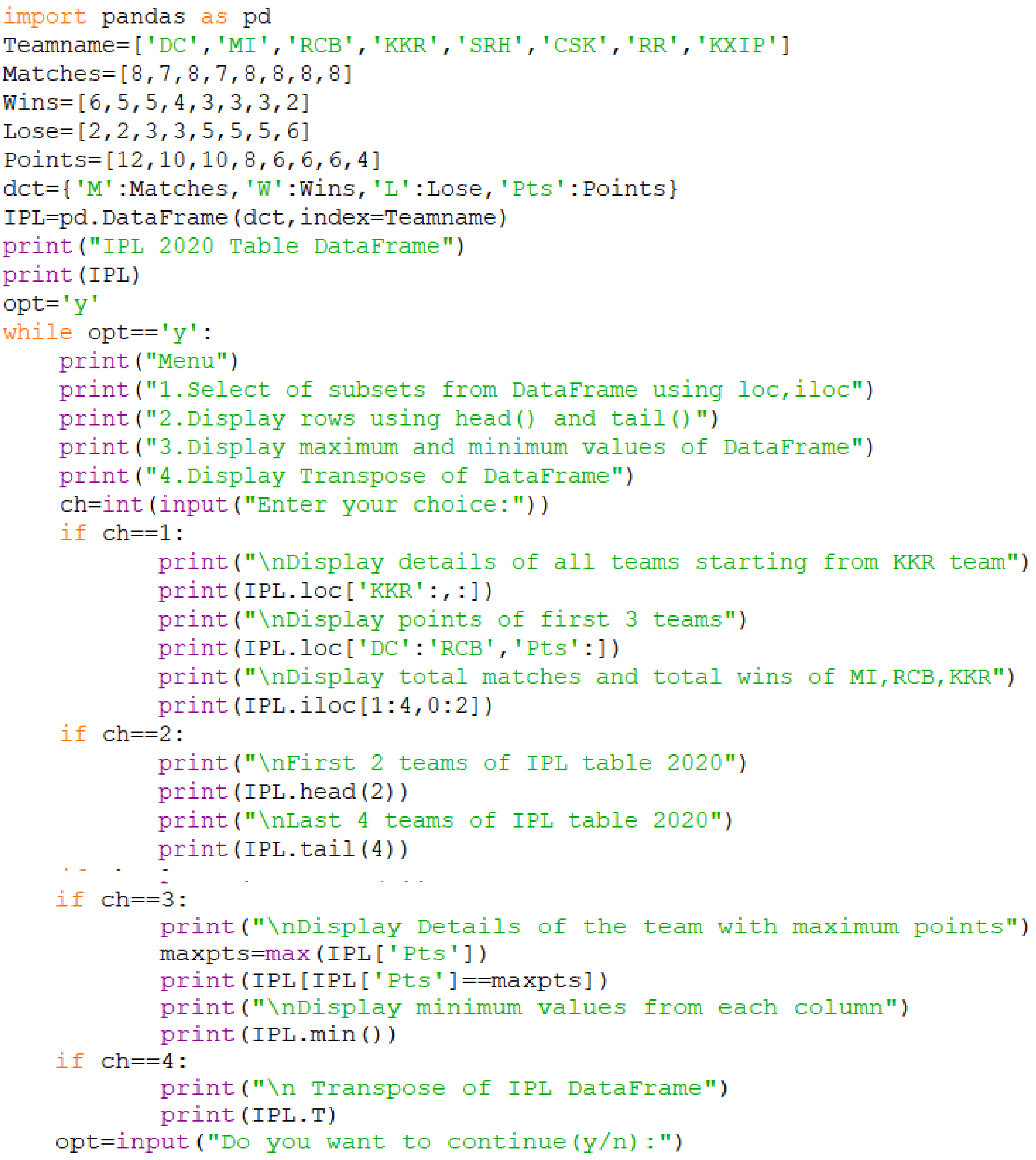
Output:



```
Initial values in dataframe
     Name  Marks  Grade
0  Preeti   34.0    NaN
1   Rohan   76.0    NaN
2   Misha   56.0    NaN
3   Sumit   91.0    NaN
4  Simran   67.0    NaN
5  Vedant   43.0    NaN

DataFrame after calculating grades
     Name  Marks Grade
0  Preeti   34.0     F
1   Rohan   76.0     A
2   Misha   56.0     C
3   Sumit   91.0    A+
4  Simran   67.0     B
5  Vedant   43.0     D

 Enter row index to display the values:1
Row value is:
Name     Rohan
Marks       76
Grade        A
Name: 1, dtype: object
Do you want to display another row values(y/n):y

 Enter row index to display the values:4
Row value is:
Name     Simran
Marks        67
Grade         B
Name: 4, dtype: object
Do you want to display another row values(y/n):n
```

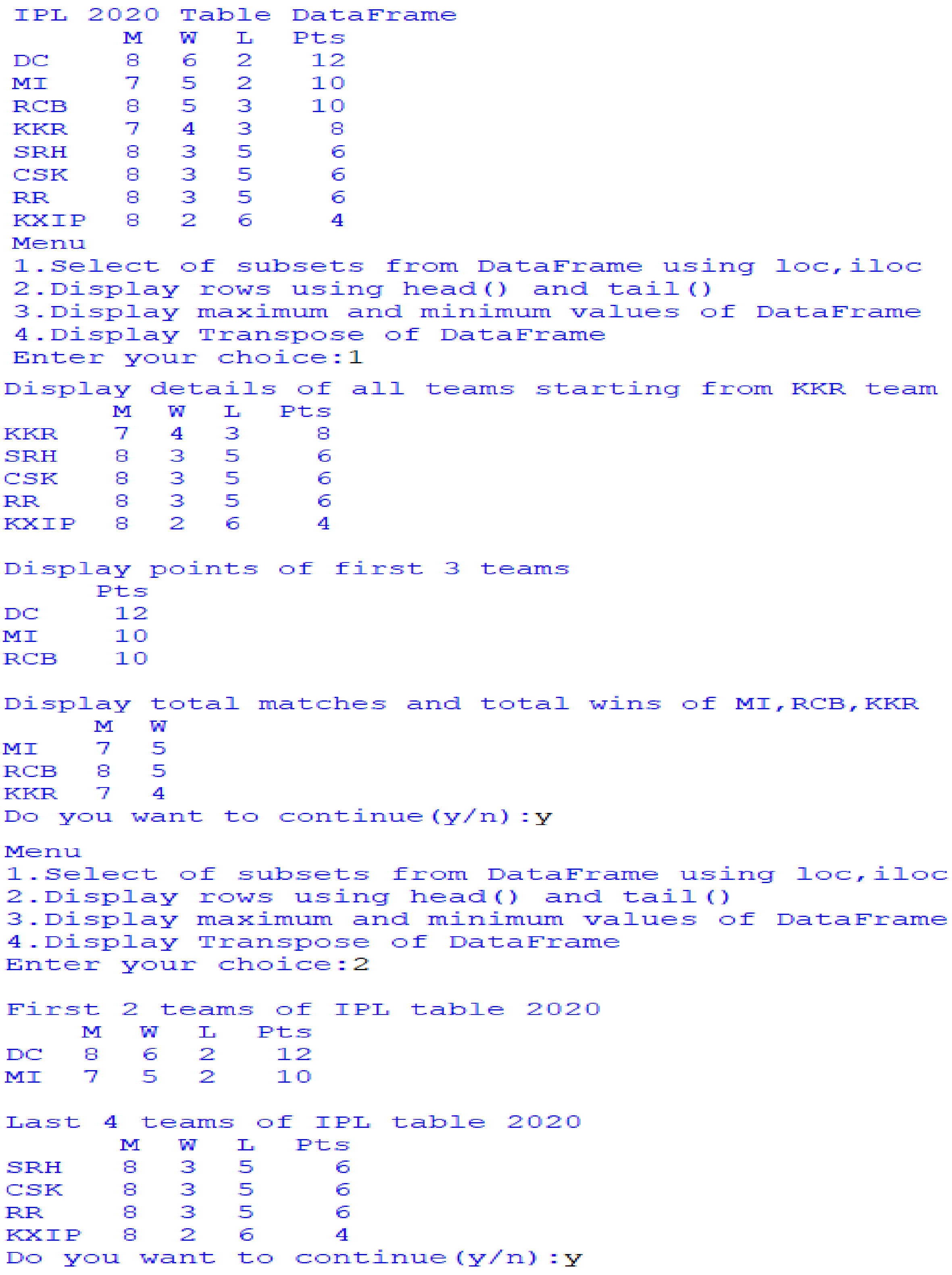
Practical 11- Write a program to perform select subsets from DataFrame using loc, iloc, head( ),tail( ) functions. Find maximum, minimum values and transpose of dataframe.

Solution:

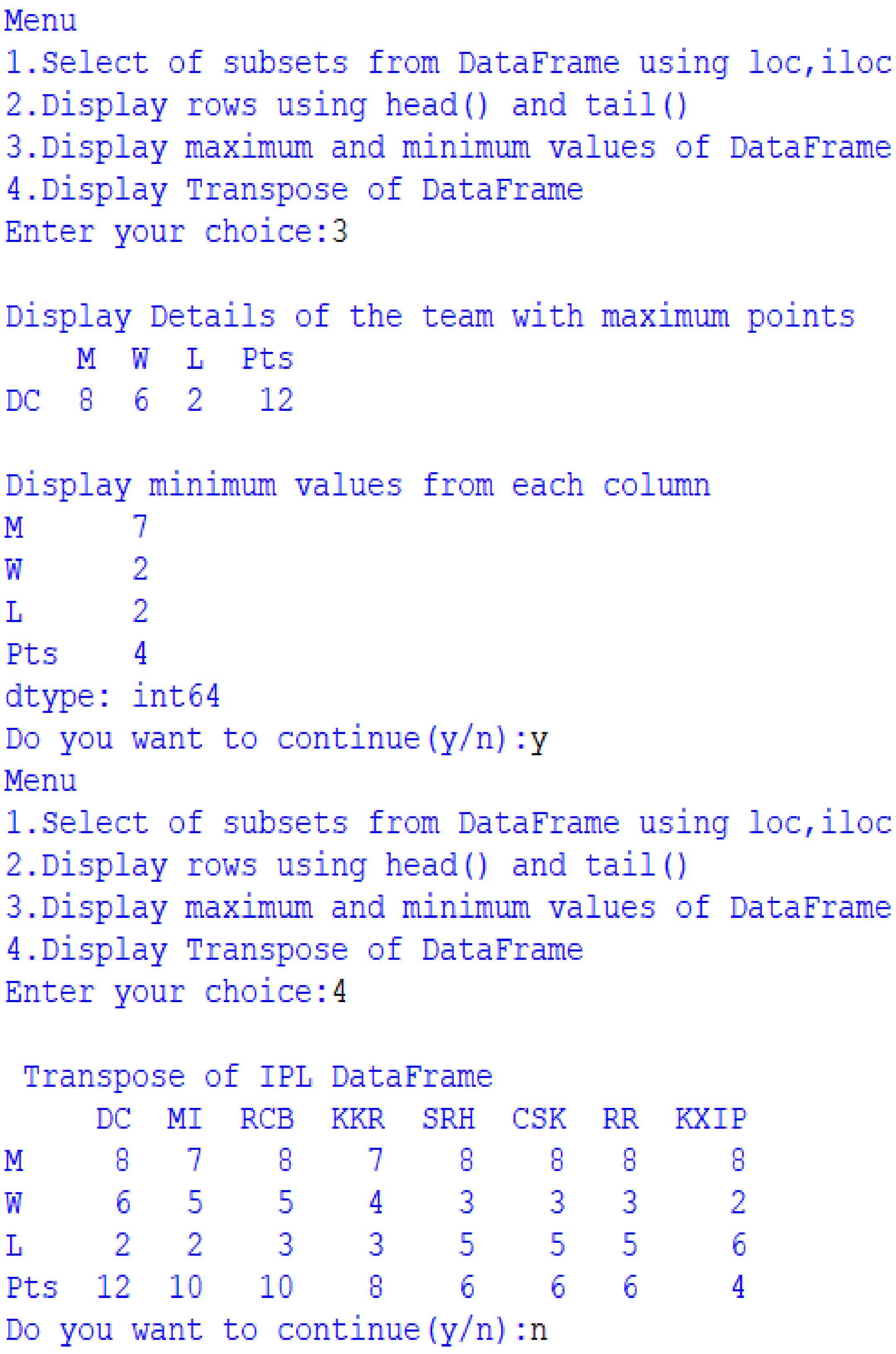


```
import pandas as pd
Teamname=['DC','MI','RCB','KKR','SRH','CSK','RR','KXIP']
Matches=[8,7,8,7,8,8,8,8]
Wins=[6,5,5,4,3,3,3,2]
Lose=[2,2,3,3,5,5,5,6]
Points=[12,10,10,8,6,6,6,4]
dct={'M':Matches,'W':Wins,'L':Lose,'Pts':Points}
IPL=pd.DataFrame(dct,index=Teamname)
print("IPL 2020 Table DataFrame")
print(IPL)
opt='y'
while opt=='y':
    print("Menu")
    print("1.Select of subsets from DataFrame using loc,iloc")
    print("2.Display rows using head() and tail()")
    print("3.Display maximum and minimum values of DataFrame")
    print("4.Display Transpose of DataFrame")
    ch=int(input("Enter your choice:"))
    if ch==1:
           print("\nDisplay details of all teams starting from KKR team")
           print(IPL.loc['KKR':,:])
           print("\nDisplay points of first 3 teams")
           print(IPL.loc['DC':'RCB','Pts':])
           print("\nDisplay total matches and total wins of MI,RCB,KKR")
           print(IPL.iloc[1:4,0:2])
    if ch==2:
           print("\nFirst 2 teams of IPL table 2020")
           print(IPL.head(2))
           print("\nLast 4 teams of IPL table 2020")
           print(IPL.tail(4))
    if ch==3:
           print("\nDisplay Details of the team with maximum points")
           maxpts=max(IPL['Pts'])
           print(IPL[IPL['Pts']==maxpts])
           print("\nDisplay minimum values from each column")
           print(IPL.min())
    if ch==4:
           print("\n Transpose of IPL DataFrame")
           print(IPL.T)
    opt=input("Do you want to continue(y/n):")
```

Output:

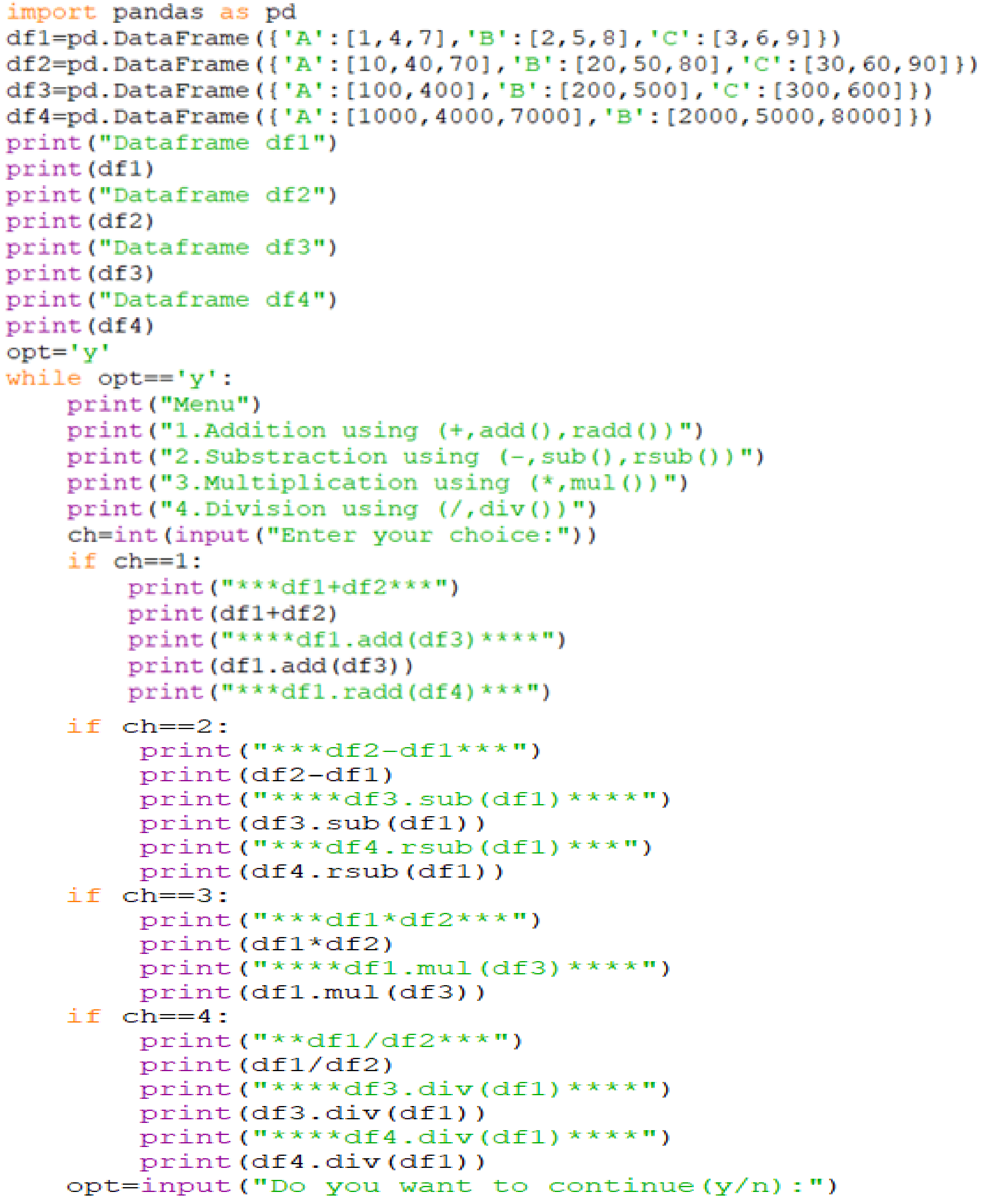


```
IPL 2020 Table DataFrame
      M  W  L  Pts
DC    8  6  2   12
MI    7  5  2   10
RCB   8  5  3   10
KKR   7  4  3    8
SRH   8  3  5    6
CSK   8  3  5    6
RR    8  3  5    6
KXIP  8  2  6    4
Menu
1.Select of subsets from DataFrame using loc,iloc
2.Display rows using head() and tail()
3.Display maximum and minimum values of DataFrame
4.Display Transpose of DataFrame
Enter your choice:1
Display details of all teams starting from KKR team
      M  W  L  Pts
KKR   7  4  3    8
SRH   8  3  5    6
CSK   8  3  5    6
RR    8  3  5    6
KXIP  8  2  6    4

Display points of first 3 teams
     Pts
DC    12
MI    10
RCB   10

Display total matches and total wins of MI,RCB,KKR
     M  W
MI   7  5
RCB  8  5
KKR  7  4
Do you want to continue(y/n):y

Menu
1.Select of subsets from DataFrame using loc,iloc
2.Display rows using head() and tail()
3.Display maximum and minimum values of DataFrame
4.Display Transpose of DataFrame
Enter your choice:2

First 2 teams of IPL table 2020
    M  W  L  Pts
DC  8  6  2   12
MI  7  5  2   10

Last 4 teams of IPL table 2020
      M  W  L  Pts
SRH   8  3  5    6
CSK   8  3  5    6
RR    8  3  5    6
KXIP  8  2  6    4
Do you want to continue(y/n):y
```



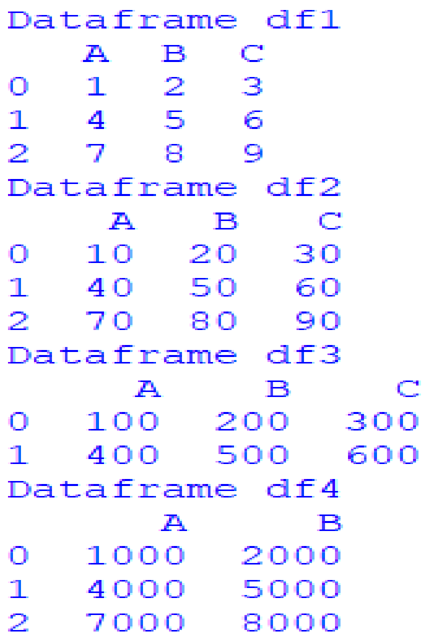
```
Menu
1.Select of subsets from DataFrame using loc,iloc
2.Display rows using head() and tail()
3.Display maximum and minimum values of DataFrame
4.Display Transpose of DataFrame
Enter your choice:3

Display Details of the team with maximum points
    M  W  L  Pts
DC  8  6  2   12

Display minimum values from each column
M      7
W      2
L      2
Pts    4
dtype: int64
Do you want to continue(y/n):y
Menu
1.Select of subsets from DataFrame using loc,iloc
2.Display rows using head() and tail()
3.Display maximum and minimum values of DataFrame
4.Display Transpose of DataFrame
Enter your choice:4

 Transpose of IPL DataFrame
     DC  MI  RCB  KKR  SRH  CSK  RR  KXIP
M     8   7    8    7    8    8   8     8
W     6   5    5    4    3    3   3     2
L     2   2    3    3    5    5   5     6
Pts  12  10   10    8    6    6   6     4
Do you want to continue(y/n):n
```

Practical 12-Write a program perform Arithmetic binary operations (Addition, Subtraction, Multiplication, Division) on dataframe.
Solution:

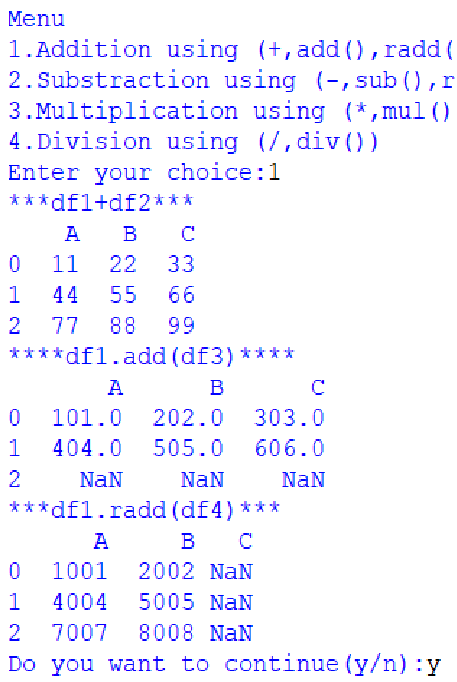


```
import pandas as pd
df1=pd.DataFrame({'A':[1,4,7],'B':[2,5,8],'C':[3,6,9]})
df2=pd.DataFrame({'A':[10,40,70],'B':[20,50,80],'C':[30,60,90]})
df3=pd.DataFrame({'A':[100,400],'B':[200,500],'C':[300,600]})
df4=pd.DataFrame({'A':[1000,4000,7000],'B':[2000,5000,8000]})
print("Dataframe df1")
print(df1)
print("Dataframe df2")
print(df2)
print("Dataframe df3")
print(df3)
print("Dataframe df4")
print(df4)
opt='y'
while opt=='y':
    print("Menu")
    print("1.Addition using (+,add(),radd())")
    print("2.Substraction using (-,sub(),rsub())")
    print("3.Multiplication using (*,mul())")
    print("4.Division using (/,div())")
    ch=int(input("Enter your choice:"))
    if ch==1:
        print("***df1+df2***")
        print(df1+df2)
        print("****df1.add(df3)****")
        print(df1.add(df3))
        print("***df1.radd(df4)***")

    if ch==2:
        print("***df2-df1***")
        print(df2-df1)
        print("****df3.sub(df1)****")
        print(df3.sub(df1))
        print("***df4.rsub(df1)***")
        print(df4.rsub(df1))
    if ch==3:
        print("***df1*df2***")
        print(df1*df2)
        print("****df1.mul(df3)****")
        print(df1.mul(df3))
    if ch==4:
        print("**df1/df2***")
        print(df1/df2)
        print("****df3.div(df1)****")
        print(df3.div(df1))
        print("****df4.div(df1)****")
        print(df4.div(df1))
    opt=input("Do you want to continue(y/n):")
```

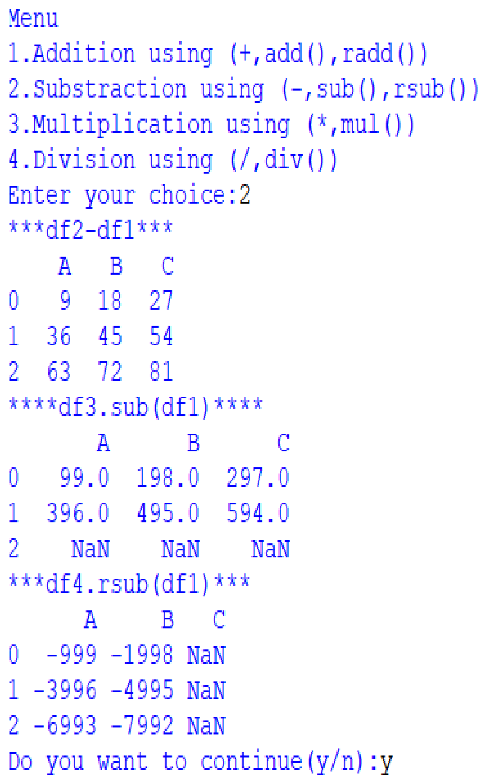
Output:



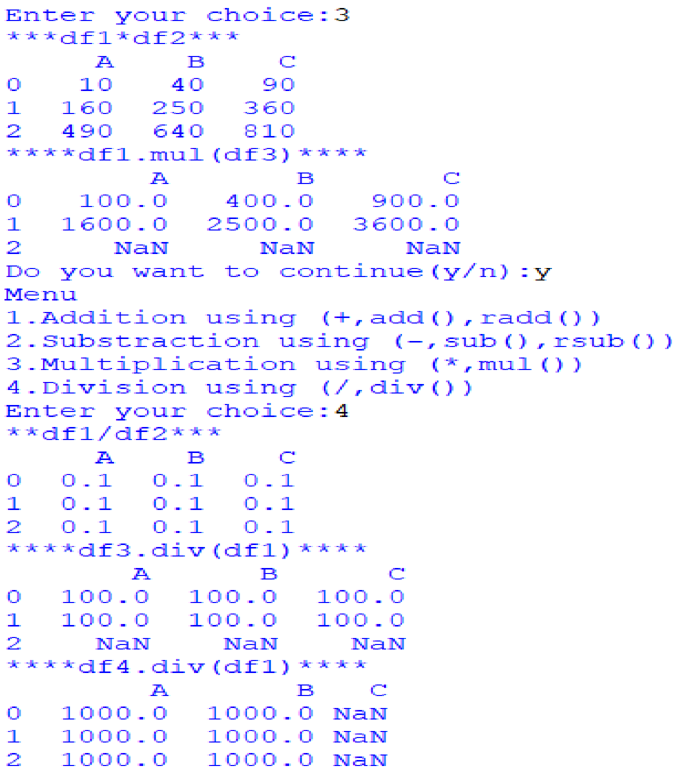
```
Dataframe df1
    A  B  C
0   1  2  3
1   4  5  6
2   7  8  9
Dataframe df2
     A   B   C
0   10  20  30
1   40  50  60
2   70  80  90
Dataframe df3
      A    B    C
0   100  200  300
1   400  500  600
Dataframe df4
       A     B
0   1000  2000
1   4000  5000
2   7000  8000
```



```
Menu
1.Addition using (+,add(),radd())
2.Substraction using (-,sub(),rsub())
3.Multiplication using (*,mul())
4.Division using (/,div())
Enter your choice:1
***df1+df2***
    A   B   C
0  11  22  33
1  44  55  66
2  77  88  99
****df1.add(df3)****
       A      B      C
0  101.0  202.0  303.0
1  404.0  505.0  606.0
2    NaN    NaN    NaN
***df1.radd(df4)***
      A     B   C
0  1001  2002 NaN
1  4004  5005 NaN
2  7007  8008 NaN
Do you want to continue(y/n):y
```



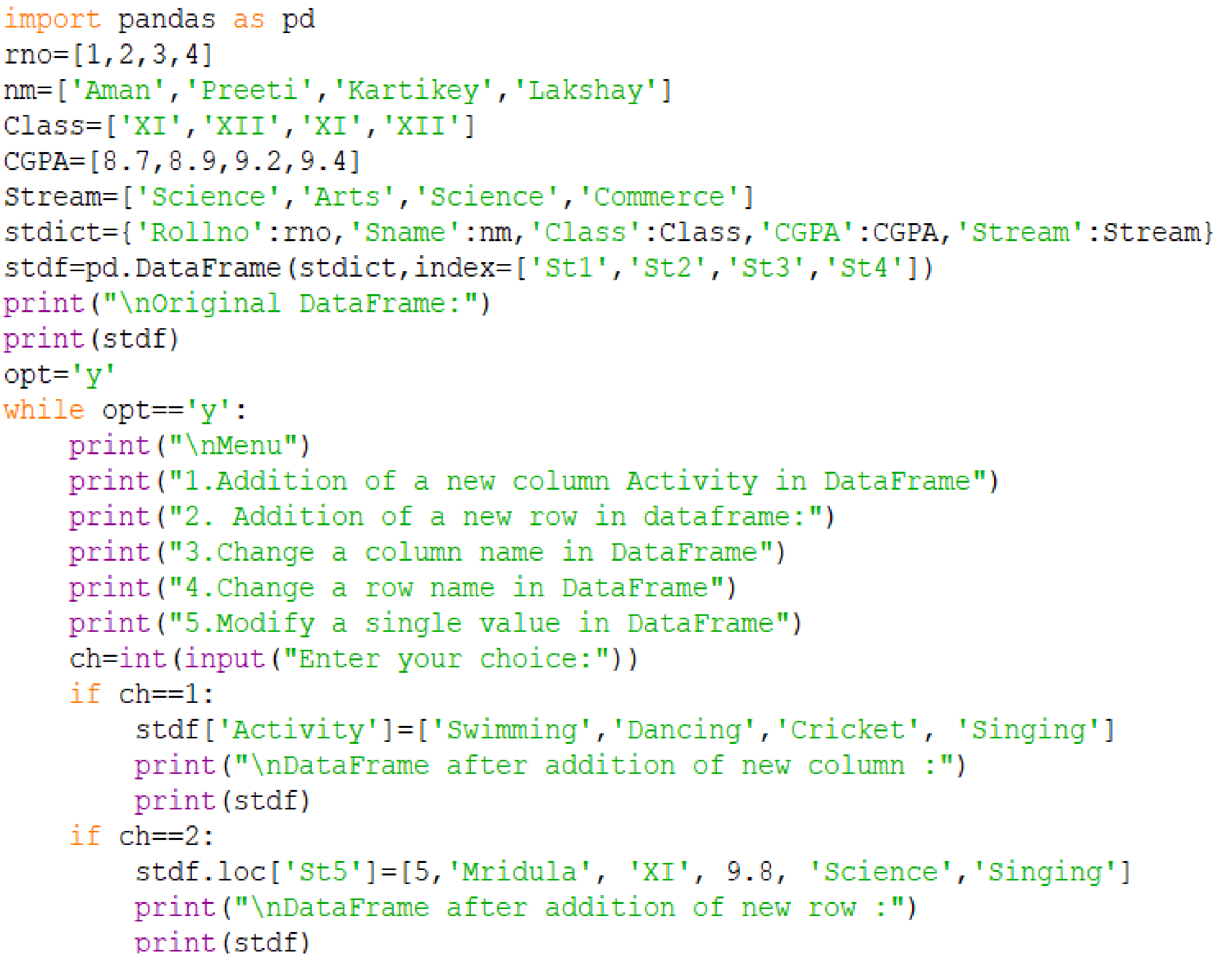
```
Menu
1.Addition using (+,add(),radd())
2.Substraction using (-,sub(),rsub())
3.Multiplication using (*,mul())
4.Division using (/,div())
Enter your choice:2
***df2-df1***
    A   B   C
0   9  18  27
1  36  45  54
2  63  72  81
****df3.sub(df1)****
       A      B      C
0   99.0  198.0  297.0
1  396.0  495.0  594.0
2    NaN    NaN    NaN
***df4.rsub(df1)***
      A     B   C
0  -999 -1998 NaN
1 -3996 -4995 NaN
2 -6993 -7992 NaN
Do you want to continue(y/n):y
```



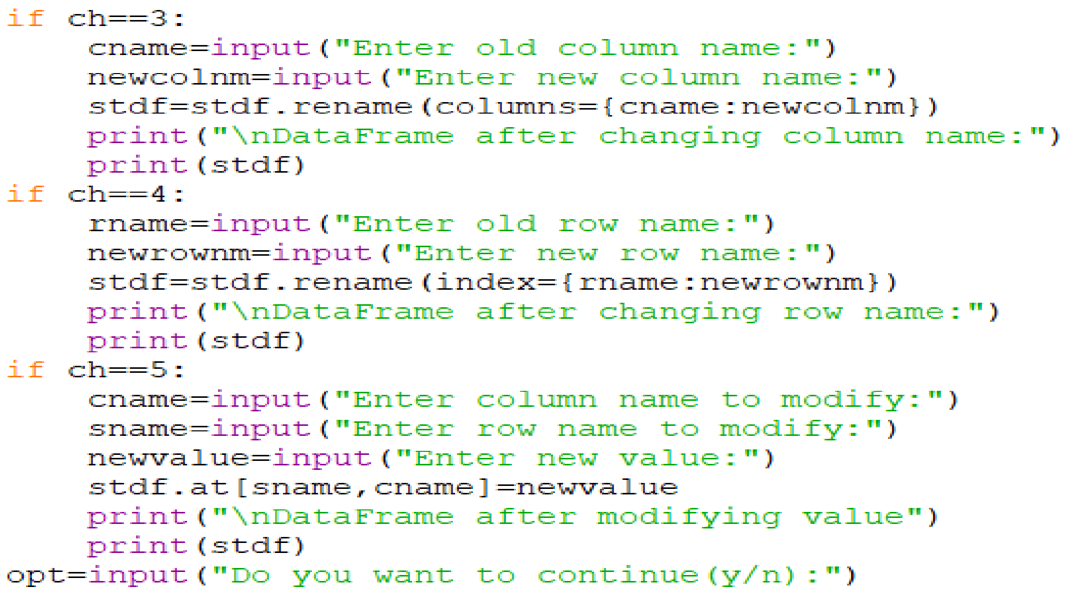
```
Enter your choice:3
***df1*df2***
     A    B    C
0   10   40   90
1  160  250  360
2  490  640  810
****df1.mul(df3)****
        A       B       C
0   100.0   400.0   900.0
1  1600.0  2500.0  3600.0
2     NaN     NaN     NaN
Do you want to continue(y/n):y
Menu
1.Addition using (+,add(),radd())
2.Substraction using (-,sub(),rsub())
3.Multiplication using (*,mul())
4.Division using (/,div())
Enter your choice:4
**df1/df2***
     A    B    C
0  0.1  0.1  0.1
1  0.1  0.1  0.1
2  0.1  0.1  0.1
****df3.div(df1)****
       A      B      C
0  100.0  100.0  100.0
1  100.0  100.0  100.0
2    NaN    NaN    NaN
****df4.div(df1)****
        A       B   C
0  1000.0  1000.0 NaN
1  1000.0  1000.0 NaN
2  1000.0  1000.0 NaN
```

Practical 13-Write a program to access and modify the values in Dataframe.

Solution:

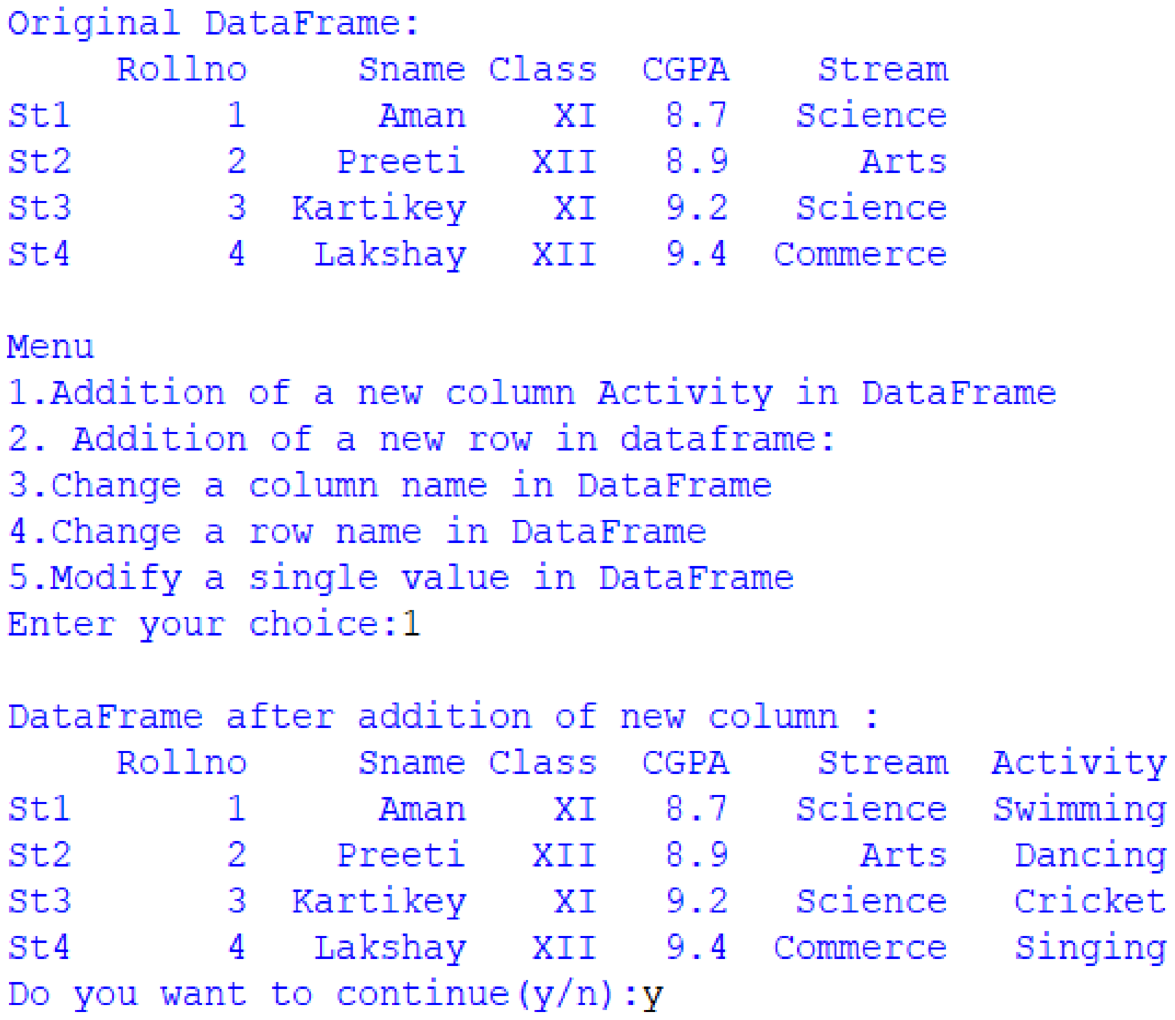


```
import pandas as pd
rno=[1,2,3,4]
nm=['Aman','Preeti','Kartikey','Lakshay']
Class=['XI','XII','XI','XII']
CGPA=[8.7,8.9,9.2,9.4]
Stream=['Science','Arts','Science','Commerce']
stdict={'Rollno':rno,'Sname':nm,'Class':Class,'CGPA':CGPA,'Stream':Stream}
stdf=pd.DataFrame(stdict,index=['St1','St2','St3','St4'])
print("\nOriginal DataFrame:")
print(stdf)
opt='y'
while opt=='y':
    print("\nMenu")
    print("1.Addition of a new column Activity in DataFrame")
    print("2. Addition of a new row in dataframe:")
    print("3.Change a column name in DataFrame")
    print("4.Change a row name in DataFrame")
    print("5.Modify a single value in DataFrame")
    ch=int(input("Enter your choice:"))
    if ch==1:
        stdf['Activity']=['Swimming','Dancing','Cricket', 'Singing']
        print("\nDataFrame after addition of new column :")
        print(stdf)
    if ch==2:
        stdf.loc['St5']=[5,'Mridula', 'XI', 9.8, 'Science','Singing']
        print("\nDataFrame after addition of new row :")
        print(stdf)
```

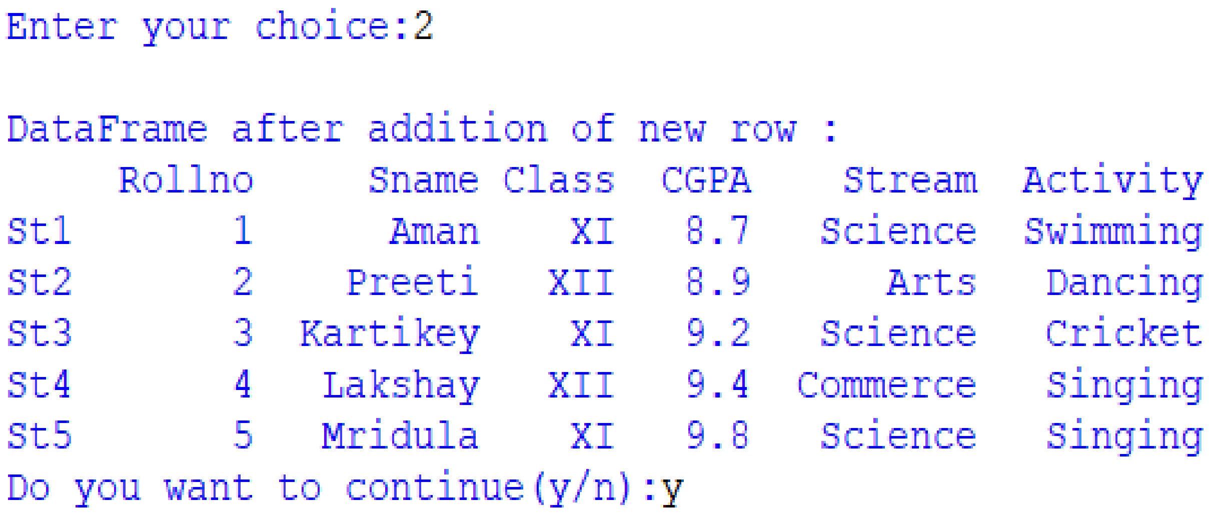


```
    if ch==3:
         cname=input("Enter old column name:")
         newcolnm=input("Enter new column name:")
         stdf=stdf.rename(columns={cname:newcolnm})
         print("\nDataFrame after changing column name:")
         print(stdf)
    if ch==4:
         rname=input("Enter old row name:")
         newrownm=input("Enter new row name:")
         stdf=stdf.rename(index={rname:newrownm})
         print("\nDataFrame after changing row name:")
         print(stdf)
    if ch==5:
         cname=input("Enter column name to modify:")
         sname=input("Enter row name to modify:")
         newvalue=input("Enter new value:")
         stdf.at[sname,cname]=newvalue
         print("\nDataFrame after modifying value")
         print(stdf)
    opt=input("Do you want to continue(y/n):")
```

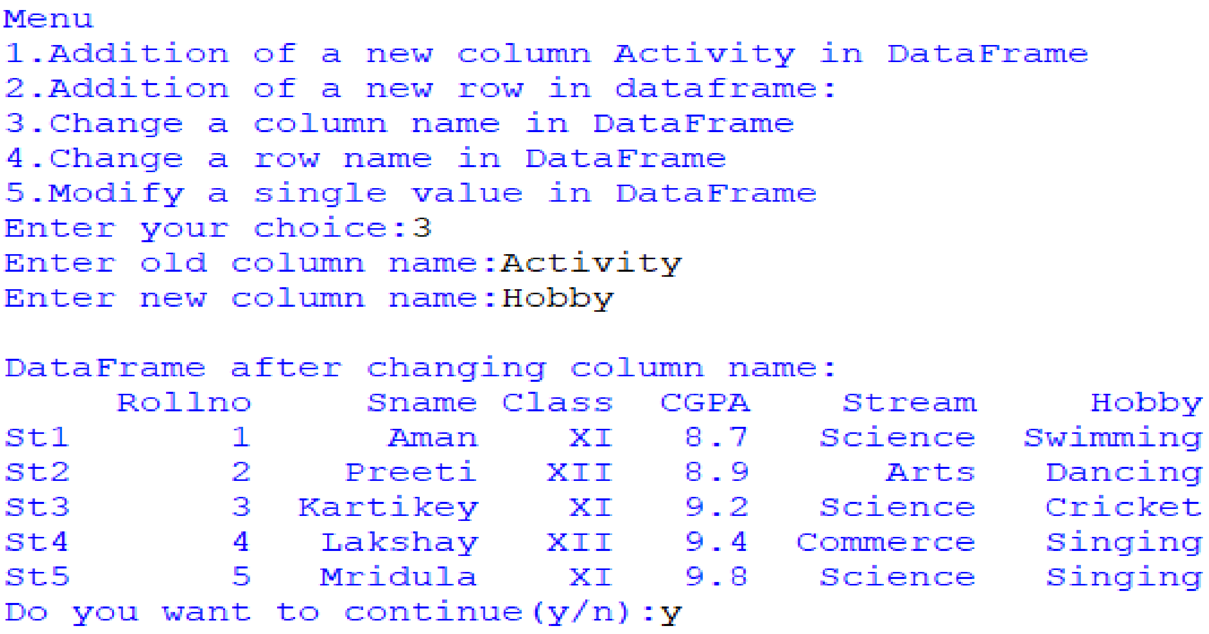
Output:



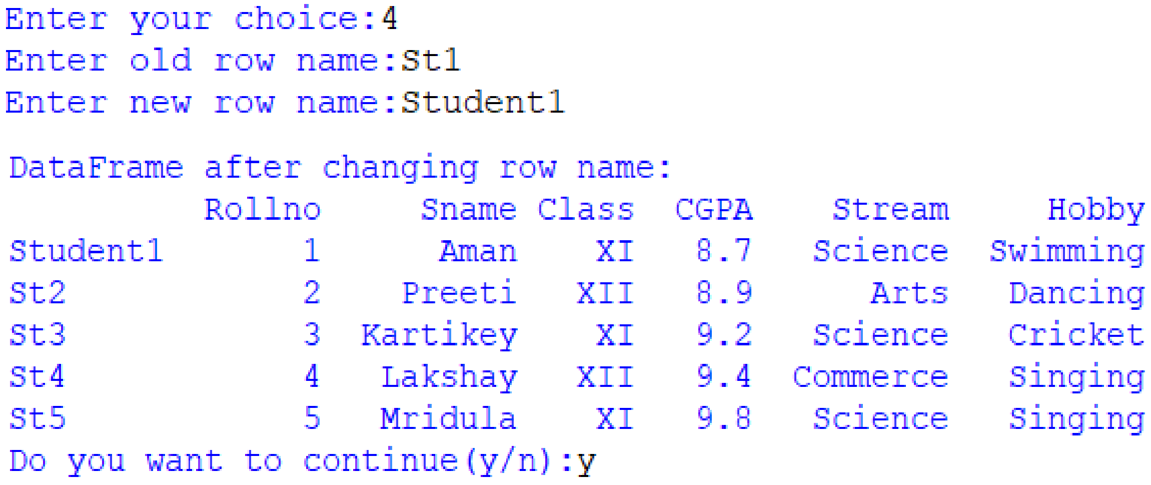
```
Original DataFrame:
     Rollno     Sname Class  CGPA    Stream
St1       1      Aman    XI   8.7   Science
St2       2    Preeti   XII   8.9      Arts
St3       3  Kartikey    XI   9.2   Science
St4       4   Lakshay   XII   9.4  Commerce

Menu
1.Addition of a new column Activity in DataFrame
2. Addition of a new row in dataframe:
3.Change a column name in DataFrame
4.Change a row name in DataFrame
5.Modify a single value in DataFrame
Enter your choice:1

DataFrame after addition of new column :
     Rollno     Sname Class  CGPA    Stream  Activity
St1       1      Aman    XI   8.7   Science  Swimming
St2       2    Preeti   XII   8.9      Arts   Dancing
St3       3  Kartikey    XI   9.2   Science   Cricket
St4       4   Lakshay   XII   9.4  Commerce   Singing
Do you want to continue(y/n):y
```



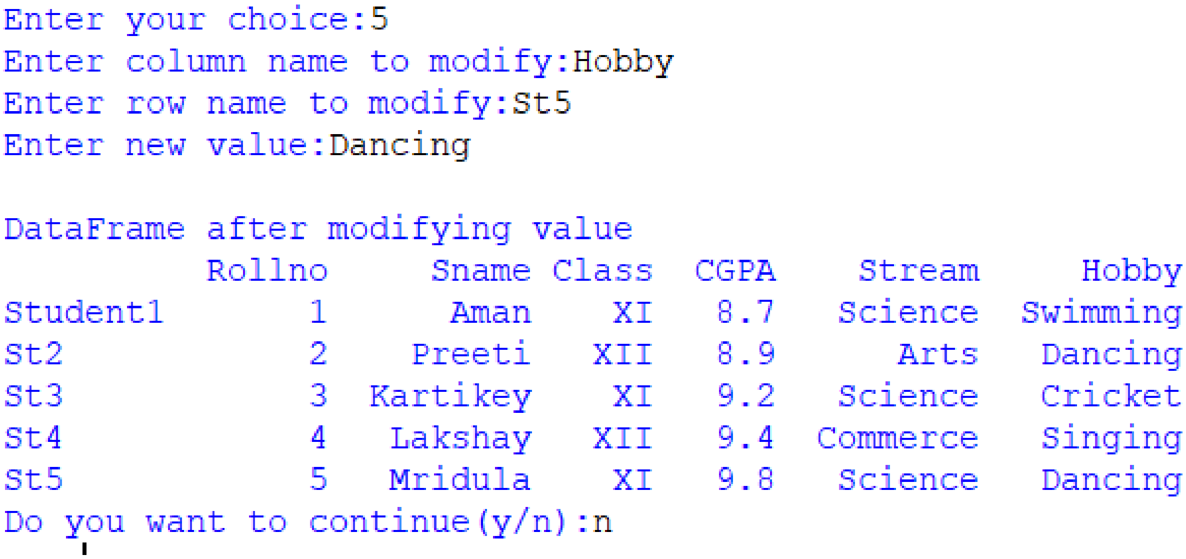
```
Enter your choice:2

DataFrame after addition of new row :
     Rollno     Sname Class  CGPA    Stream  Activity
St1       1      Aman    XI   8.7   Science  Swimming
St2       2    Preeti   XII   8.9      Arts   Dancing
St3       3  Kartikey    XI   9.2   Science   Cricket
St4       4   Lakshay   XII   9.4  Commerce   Singing
St5       5   Mridula    XI   9.8   Science   Singing
Do you want to continue(y/n):y
```



```
Menu
1.Addition of a new column Activity in DataFrame
2.Addition of a new row in dataframe:
3.Change a column name in DataFrame
4.Change a row name in DataFrame
5.Modify a single value in DataFrame
Enter your choice:3
Enter old column name:Activity
Enter new column name:Hobby

DataFrame after changing column name:
     Rollno     Sname Class  CGPA    Stream     Hobby
St1       1      Aman    XI   8.7   Science  Swimming
St2       2    Preeti   XII   8.9      Arts   Dancing
St3       3  Kartikey    XI   9.2   Science   Cricket
St4       4   Lakshay   XII   9.4  Commerce   Singing
St5       5   Mridula    XI   9.8   Science   Singing
Do you want to continue(y/n):y
```

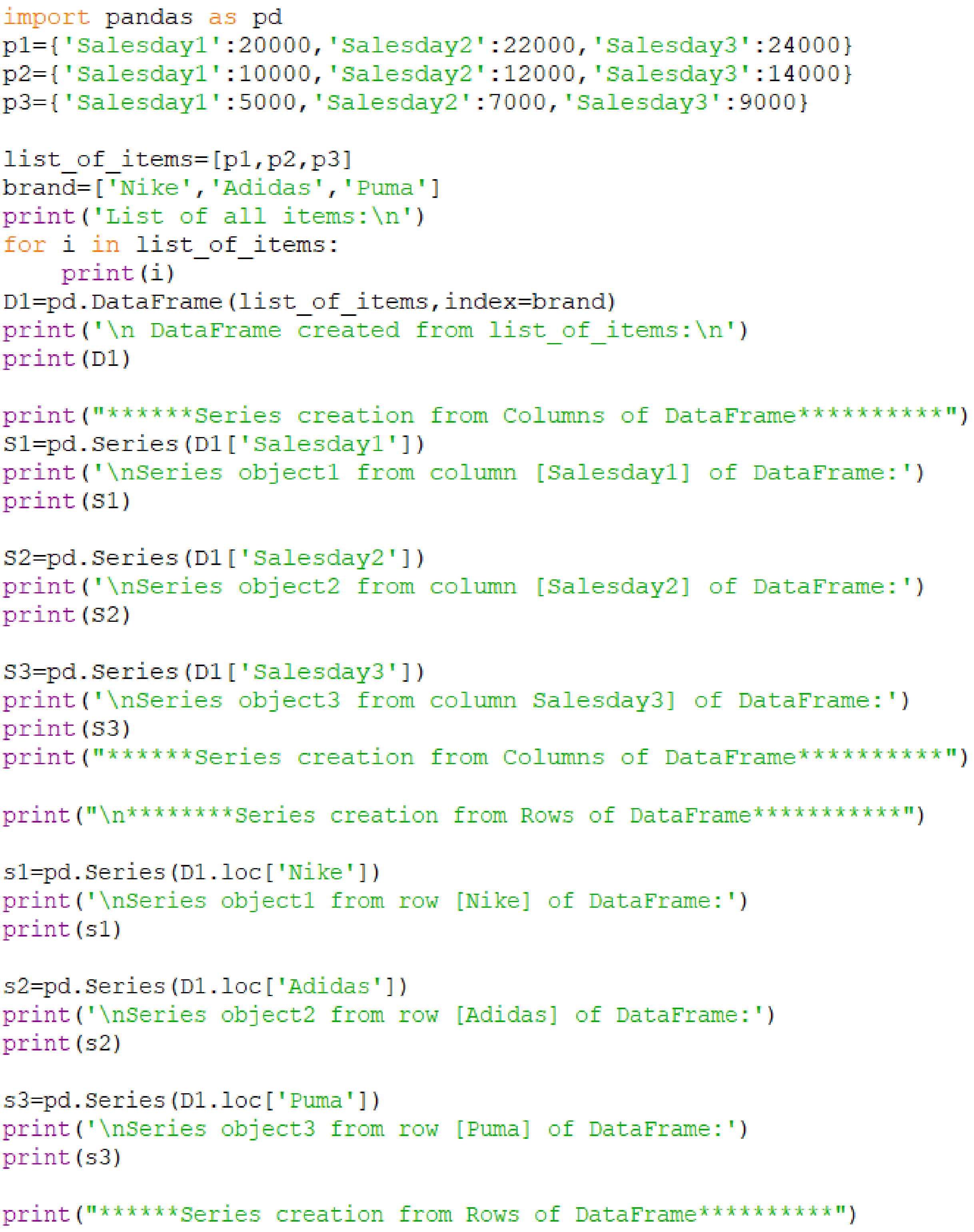


```
Enter your choice:4
Enter old row name:St1
Enter new row name:Student1

DataFrame after changing row name:
          Rollno     Sname Class  CGPA    Stream     Hobby
Student1       1      Aman    XI   8.7   Science  Swimming
St2            2    Preeti   XII   8.9      Arts   Dancing
St3            3  Kartikey    XI   9.2   Science   Cricket
St4            4   Lakshay   XII   9.4  Commerce   Singing
St5            5   Mridula    XI   9.8   Science   Singing
Do you want to continue(y/n):y
```



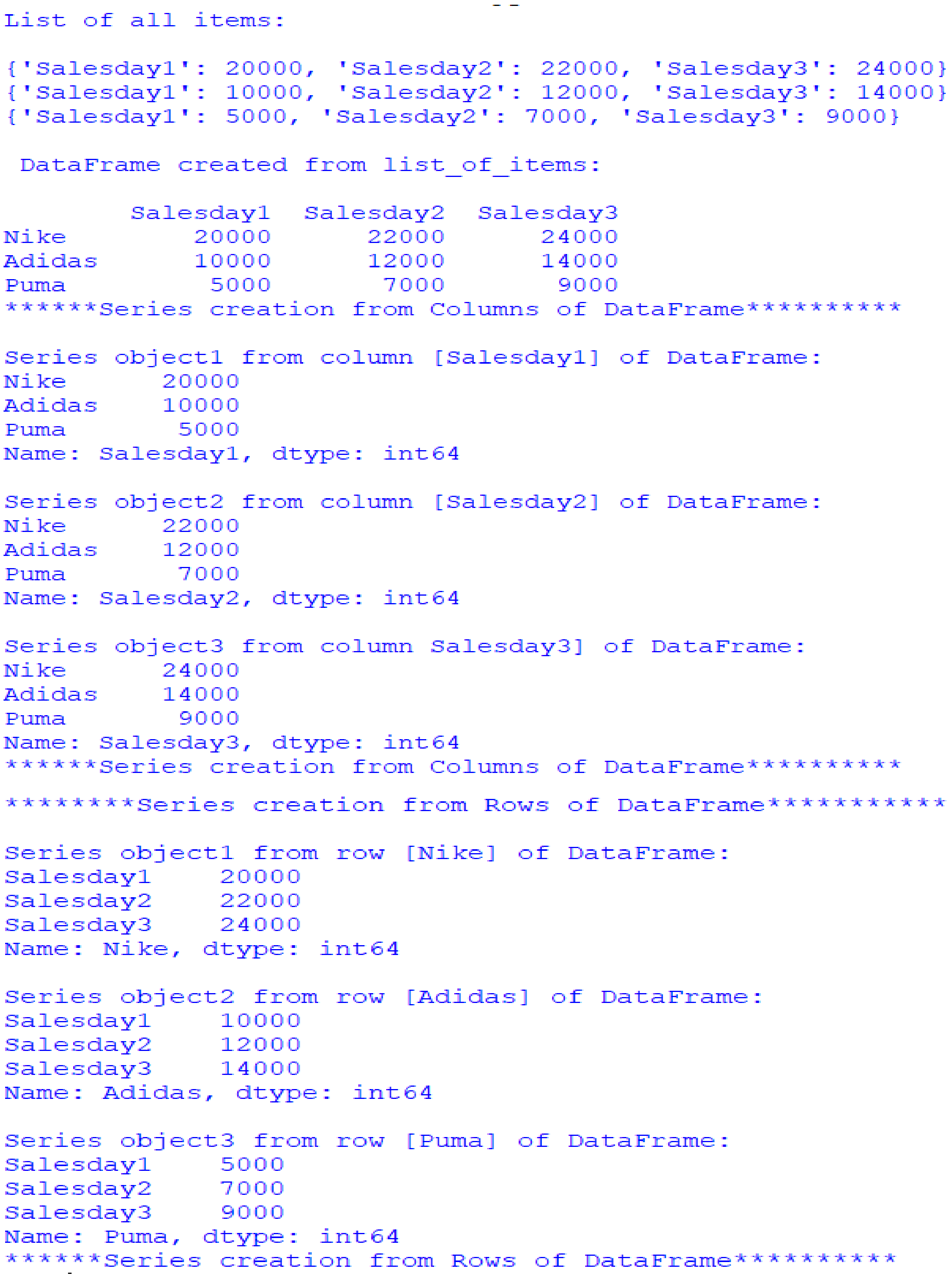
```
Enter your choice:5
Enter column name to modify:Hobby
Enter row name to modify:St5
Enter new value:Dancing

DataFrame after modifying value
          Rollno     Sname Class  CGPA    Stream     Hobby
Student1       1      Aman    XI   8.7   Science  Swimming
St2            2    Preeti   XII   8.9      Arts   Dancing
St3            3  Kartikey    XI   9.2   Science   Cricket
St4            4   Lakshay   XII   9.4  Commerce   Singing
St5            5   Mridula    XI   9.8   Science   Dancing
Do you want to continue(y/n):n
```

Program 14- Program for creating Series objects from Dataframe columns and DataFrame rows.
Solution:



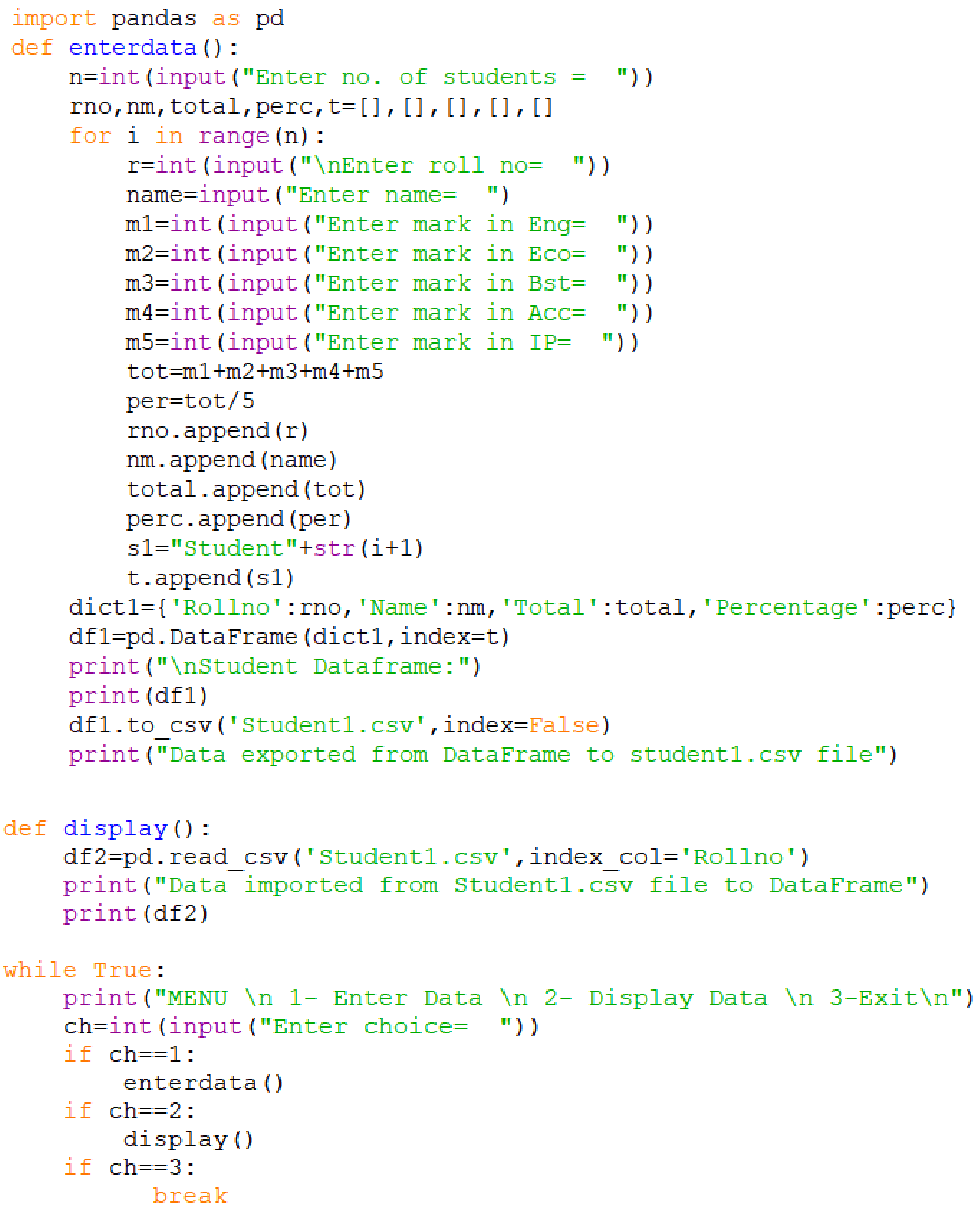
```
import pandas as pd
p1={'Salesday1':20000,'Salesday2':22000,'Salesday3':24000}
p2={'Salesday1':10000,'Salesday2':12000,'Salesday3':14000}
p3={'Salesday1':5000,'Salesday2':7000,'Salesday3':9000}

list_of_items=[p1,p2,p3]
brand=['Nike','Adidas','Puma']
print('List of all items:\n')
for i in list_of_items:
    print(i)
D1=pd.DataFrame(list_of_items,index=brand)
print('\n DataFrame created from list_of_items:\n')
print(D1)

print("******Series creation from Columns of DataFrame**********")
S1=pd.Series(D1['Salesday1'])
print('\nSeries object1 from column [Salesday1] of DataFrame:')
print(S1)

S2=pd.Series(D1['Salesday2'])
print('\nSeries object2 from column [Salesday2] of DataFrame:')
print(S2)

S3=pd.Series(D1['Salesday3'])
print('\nSeries object3 from column Salesday3] of DataFrame:')
print(S3)
print("******Series creation from Columns of DataFrame**********")

print("\n********Series creation from Rows of DataFrame***********")

s1=pd.Series(D1.loc['Nike'])
print('\nSeries object1 from row [Nike] of DataFrame:')
print(s1)

s2=pd.Series(D1.loc['Adidas'])
print('\nSeries object2 from row [Adidas] of DataFrame:')
print(s2)

s3=pd.Series(D1.loc['Puma'])
print('\nSeries object3 from row [Puma] of DataFrame:')
print(s3)

print("******Series creation from Rows of DataFrame**********")
```

Output:



```
List of all items:

{'Salesday1': 20000, 'Salesday2': 22000, 'Salesday3': 24000}
{'Salesday1': 10000, 'Salesday2': 12000, 'Salesday3': 14000}
{'Salesday1': 5000, 'Salesday2': 7000, 'Salesday3': 9000}

 DataFrame created from list_of_items:

        Salesday1  Salesday2  Salesday3
Nike        20000      22000      24000
Adidas      10000      12000      14000
Puma         5000       7000       9000
******Series creation from Columns of DataFrame**********

Series object1 from column [Salesday1] of DataFrame:
Nike      20000
Adidas    10000
Puma       5000
Name: Salesday1, dtype: int64

Series object2 from column [Salesday2] of DataFrame:
Nike      22000
Adidas    12000
Puma       7000
Name: Salesday2, dtype: int64

Series object3 from column Salesday3] of DataFrame:
Nike      24000
Adidas    14000
Puma       9000
Name: Salesday3, dtype: int64
******Series creation from Columns of DataFrame**********

********Series creation from Rows of DataFrame***********

Series object1 from row [Nike] of DataFrame:
Salesday1    20000
Salesday2    22000
Salesday3    24000
Name: Nike, dtype: int64

Series object2 from row [Adidas] of DataFrame:
Salesday1    10000
Salesday2    12000
Salesday3    14000
Name: Adidas, dtype: int64

Series object3 from row [Puma] of DataFrame:
Salesday1    5000
Salesday2    7000
Salesday3    9000
Name: Puma, dtype: int64
******Series creation from Rows of DataFrame**********
```

Practical 15-Write a program to create a DataFrame for student details and create a CSV file using this DataFrame and display contents of CSV file.
Solution:



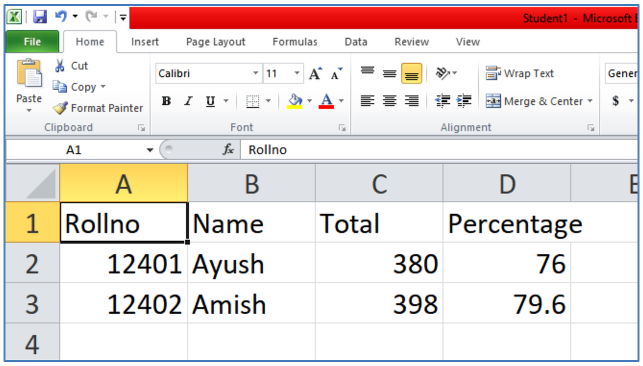
```
import pandas as pd
def enterdata():
    n=int(input("Enter no. of students =  "))
    rno,nm,total,perc,t=[],[],[],[],[]
    for i in range(n):
        r=int(input("\nEnter roll no=  "))
        name=input("Enter name=  ")
        m1=int(input("Enter mark in Eng=  "))
        m2=int(input("Enter mark in Eco=  "))
        m3=int(input("Enter mark in Bst=  "))
        m4=int(input("Enter mark in Acc=  "))
        m5=int(input("Enter mark in IP=  "))
        tot=m1+m2+m3+m4+m5
        per=tot/5
        rno.append(r)
        nm.append(name)
        total.append(tot)
        perc.append(per)
        s1="Student"+str(i+1)
        t.append(s1)
    dict1={'Rollno':rno,'Name':nm,'Total':total,'Percentage':perc}
    df1=pd.DataFrame(dict1,index=t)
    print("\nStudent Dataframe:")
    print(df1)
    df1.to_csv('Student1.csv',index=False)
    print("Data exported from DataFrame to student1.csv file")


def display():
    df2=pd.read_csv('Student1.csv',index_col='Rollno')
    print("Data imported from Student1.csv file to DataFrame")
    print(df2)

while True:
    print("MENU \n 1- Enter Data \n 2- Display Data \n 3-Exit\n")
    ch=int(input("Enter choice=  "))
    if ch==1:
        enterdata()
    if ch==2:
        display()
    if ch==3:
           break
```

Output:

```
MENU
 1- Enter Data
 2- Display Data
 3-Exit

Enter choice=  1
Enter no. of students =  2

Enter roll no=  12401
Enter name=  Ayush
Enter mark in Eng=  70
Enter mark in Eco=  78
Enter mark in Bst=  77
Enter mark in Acc=  80
Enter mark in IP=  75

Enter roll no=  12402
Enter name=  Amish
Enter mark in Eng=  80
Enter mark in Eco=  78
Enter mark in Bst=  74
Enter mark in Acc=  86
Enter mark in IP=  80

Student Dataframe:
          Rollno  Name  Total  Percentage
Student1   12401  Ayush   380        76.0
Student2   12402  Amish   398        79.6
Data exported from DataFrame to student1.csv file

 MENU
  1- Enter Data
  2- Display Data
  3-Exit

 Enter choice=  2
 Data imported from Student1.csv file to DataFrame
          Name  Total  Percentage
 Rollno
 12401   Ayush   380        76.0
 12402   Amish   398        79.6
```

Contents of Student1.csv file

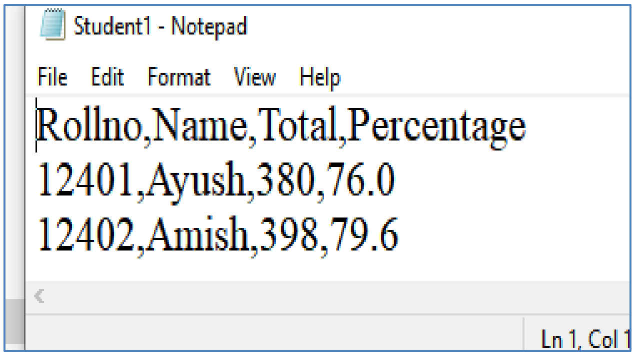
1) Open with excel



Rollno

	A	B	C	D	E
1	Rollno	Name	Total	Percentage	
2	12401	Ayush	380	76	
3	12402	Amish	398	79.6	
4					

2) Open with Notepad

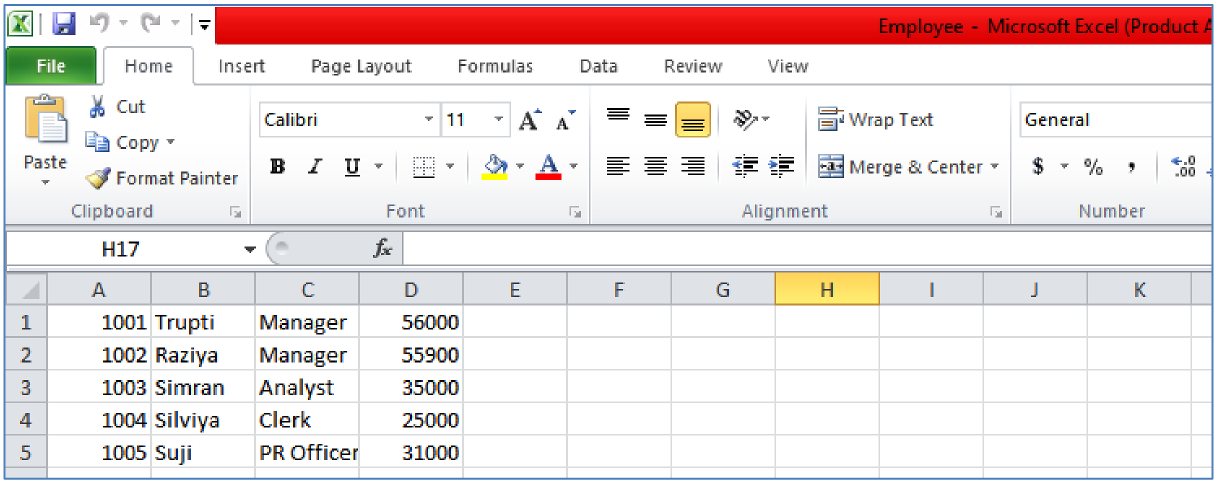


```
Rollno,Name,Total,Percentage
12401,Ayush,380,76.0
12402,Amish,398,79.6
```

Practical 16- Write a program to read a CSV file and display contents in different format. Copy the CSV file into other CSV file with different separator.

Solution:

First create a CSV file Employee.csv



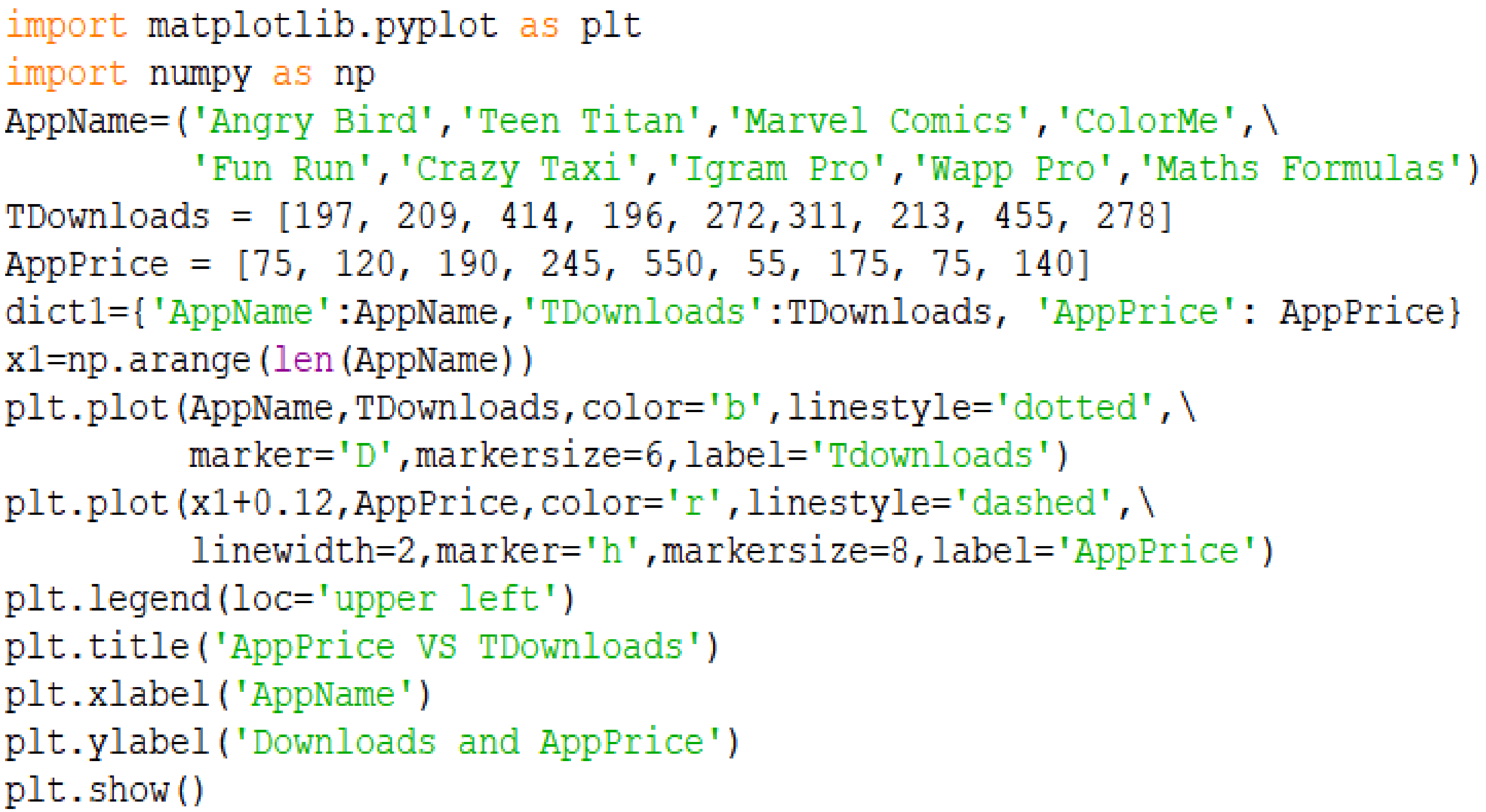
	A	B	C	D	E	F	G	H	I	J	K
1	1001	Trupti	Manager	56000							
2	1002	Raziya	Manager	55900							
3	1003	Simran	Analyst	35000							
4	1004	Silviya	Clerk	25000							
5	1005	Suji	PR Officer	31000							

```
import pandas as pd
df1=pd.read_csv("Employee.csv",header=None)
rows,columns=df1.shape
print("No. of rows in Employee.csv file are:",rows)
print("No. of columns in Employee.csv file are:",columns)
print("The contents of Employee.csv file without any column header:")
print(df1)

print("\nThe contents of CSV file by specifying own column names:")
df2=pd.read_csv("Employee.csv",names=['EmpNo','EmpName','Designation','Salary'])
print(df2)

print("\nDisplay middle row of csv file:")
df3=pd.read_csv("Employee.csv",names=['EmpNo','EmpName','Designation','Salary'],nrows=3)
print(df3.tail(1))

print("\nThe contents of CSV file CSV file by skiping 1st and 3rd row index:")
df4=pd.read_csv("Employee.csv",names=['EmpNo','EmpName','Designation','Salary'],skiprows=[1,3])
print(df4)

print("\nGetting dataframe Index from CSV file:")
df5=pd.read_csv("Employee.csv",names=['EmpNo','EmpName','Designation','Salary'],index_col="EmpNo")
print(df5)

print("\nTotal salary of all employees:",df1[3].sum())
print("Maximum salary:",df2.Salary.max())
print("Minimum salary:",df2.Salary.min())

df2.to_csv("Employee1.csv",sep='|',index=False)
df6=pd.read_csv("Employee1.csv")
print("\nThe contents of CSV file transferred to other CSV file with seperator='|'")
print(df6)
```

Output:

```
No. of rows in Employee.csv file are: 5
No. of columns in Employee.csv file are: 4
The contents of Employee.csv file without any column header:
      0        1          2      3
0  1001   Trupti     Manager  56000
1  1002   Raziya     Manager  55900
2  1003   Simran     Analyst  35000
3  1004  Silviya       Clerk  25000
4  1005     Suji  PR Officer  31000

The contents of CSV file by specifying own column names:
   EmpNo  EmpName Designation  Salary
0   1001   Trupti     Manager   56000
1   1002   Raziya     Manager   55900
2   1003   Simran     Analyst   35000
3   1004  Silviya       Clerk   25000
4   1005     Suji  PR Officer   31000

Display middle row of csv file:
   EmpNo EmpName Designation  Salary
2   1003  Simran     Analyst   35000

The contents of CSV file CSV file by skiping 1st and 3rd row index:
   EmpNo EmpName Designation  Salary
0   1001  Trupti     Manager   56000
1   1003  Simran     Analyst   35000
2   1005    Suji  PR Officer   31000


Getting dataframe Index from CSV file:
       EmpName Designation  Salary
EmpNo
1001    Trupti     Manager   56000
1002    Raziya     Manager   55900
1003    Simran     Analyst   35000
1004   Silviya       Clerk   25000
1005      Suji  PR Officer   31000

Total salary of all employees: 202900
Maximum salary: 56000
Minimum salary: 25000

The contents of CSV file transferred to other CSV file with seperator='|'
  EmpNo|EmpName|Designation|Salary
0        1001|Trupti|Manager|56000
1        1002|Raziya|Manager|55900
2        1003|Simran|Analyst|35000
3         1004|Silviya|Clerk|25000
4       1005|Suji|PR Officer|31000
```

Practical 17-Write a program to draw line chart depicting the prices of the apps and download of the apps.
Solution:



```
import matplotlib.pyplot as plt
import numpy as np
AppName=('Angry Bird','Teen Titan','Marvel Comics','ColorMe',\
         'Fun Run','Crazy Taxi','Igram Pro','Wapp Pro','Maths Formulas')
TDownloads = [197, 209, 414, 196, 272,311, 213, 455, 278]
AppPrice = [75, 120, 190, 245, 550, 55, 175, 75, 140]
dict1={'AppName':AppName,'TDownloads':TDownloads, 'AppPrice': AppPrice}
x1=np.arange(len(AppName))
plt.plot(AppName,TDownloads,color='b',linestyle='dotted',\
         marker='D',markersize=6,label='Tdownloads')
plt.plot(x1+0.12,AppPrice,color='r',linestyle='dashed',\
         linewidth=2,marker='h',markersize=8,label='AppPrice')
plt.legend(loc='upper left')
plt.title('AppPrice VS TDownloads')
plt.xlabel('AppName')
plt.ylabel('Downloads and AppPrice')
plt.show()
```

Output:

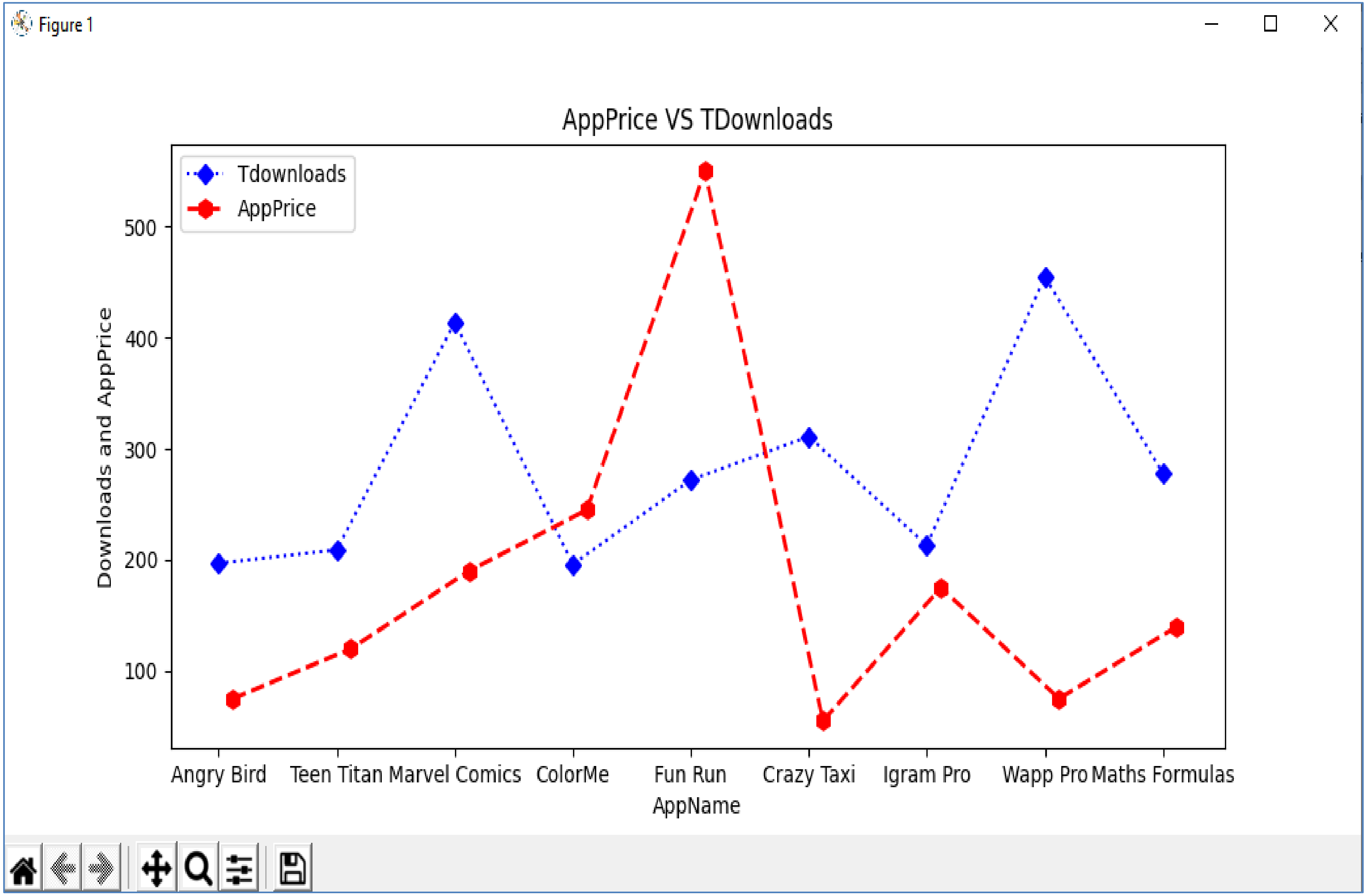
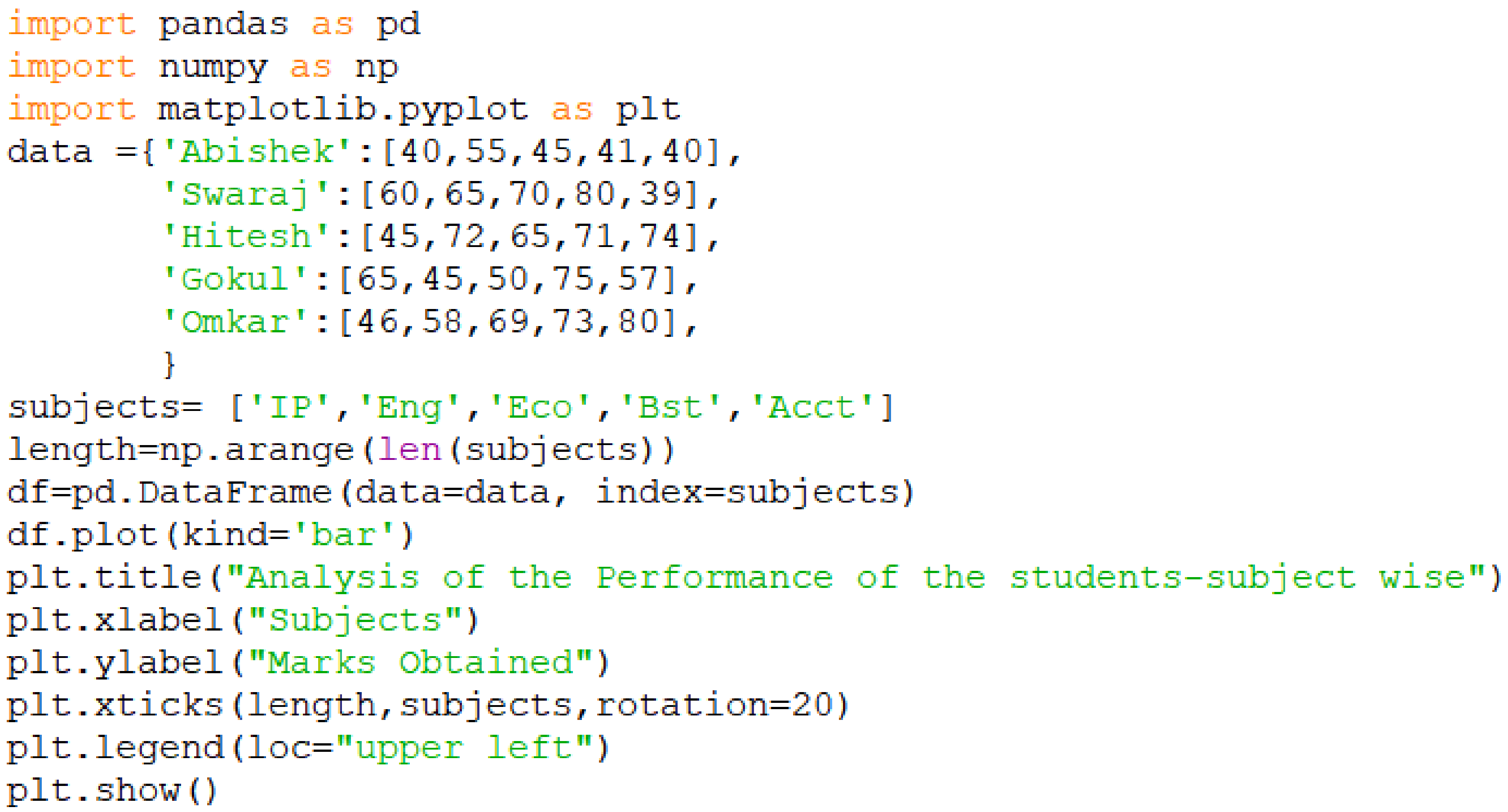


Figure 1
AppPrice VS TDownloads
Tdownloads
AppPrice
Downloads and AppPrice
500
400
300
200
100
Angry Bird
Teen Titan
Marvel Comics
ColorMe
Fun Run
Crazy Taxi
Igram Pro
Wapp Pro
Maths Formulas
AppName

Practical 18-Given the school result data, Plot bar graph using function of Dataframe for subject wise analysis of performance of the students.

Solution:



```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
data ={'Abishek':[40,55,45,41,40],
       'Swaraj':[60,65,70,80,39],
       'Hitesh':[45,72,65,71,74],
       'Gokul':[65,45,50,75,57],
       'Omkar':[46,58,69,73,80],
       }
subjects= ['IP','Eng','Eco','Bst','Acct']
length=np.arange(len(subjects))
df=pd.DataFrame(data=data, index=subjects)
df.plot(kind='bar')
plt.title("Analysis of the Performance of the students-subject wise")
plt.xlabel("Subjects")
plt.ylabel("Marks Obtained")
plt.xticks(length,subjects,rotation=20)
plt.legend(loc="upper left")
plt.show()
```

Output:

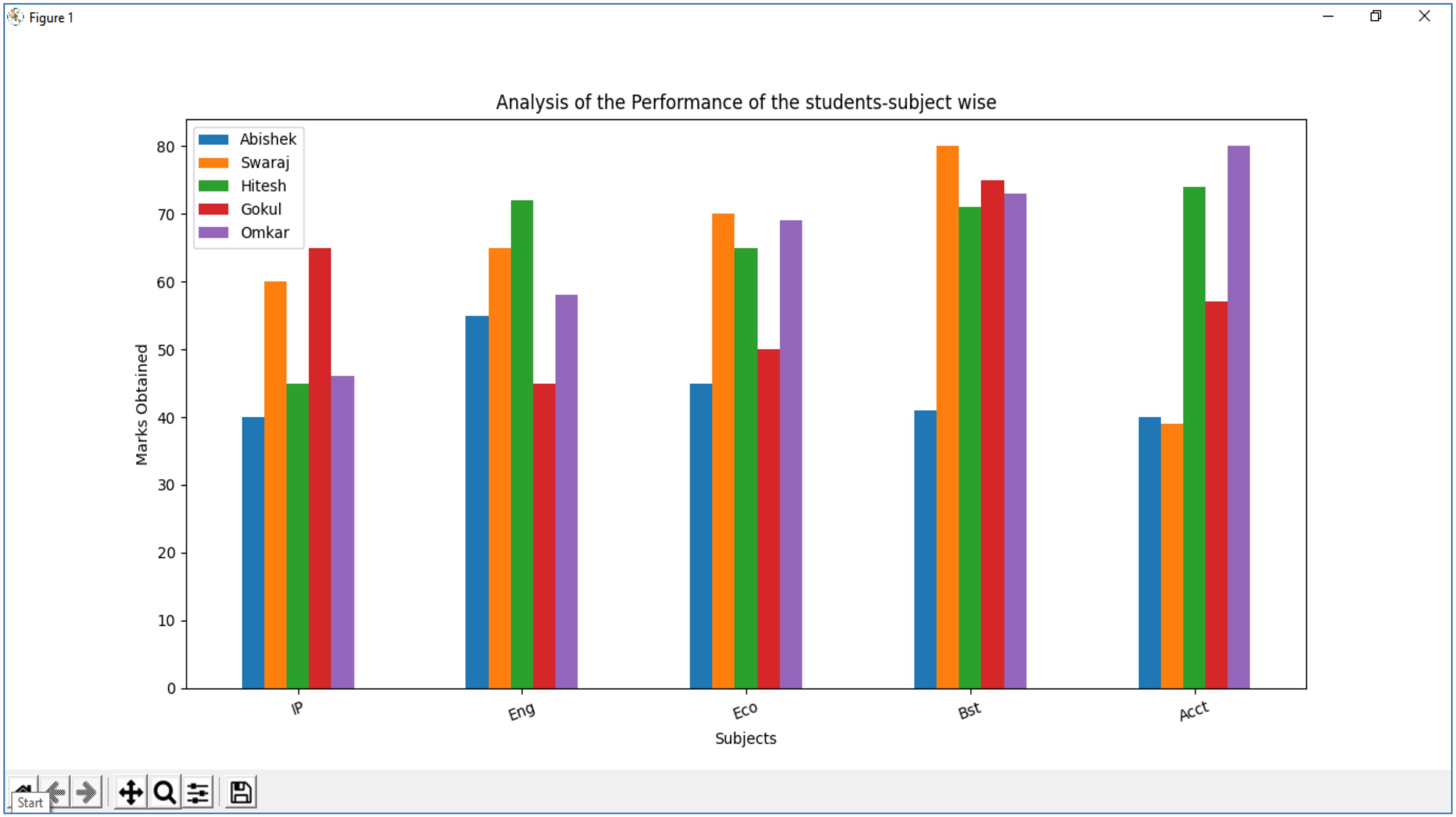
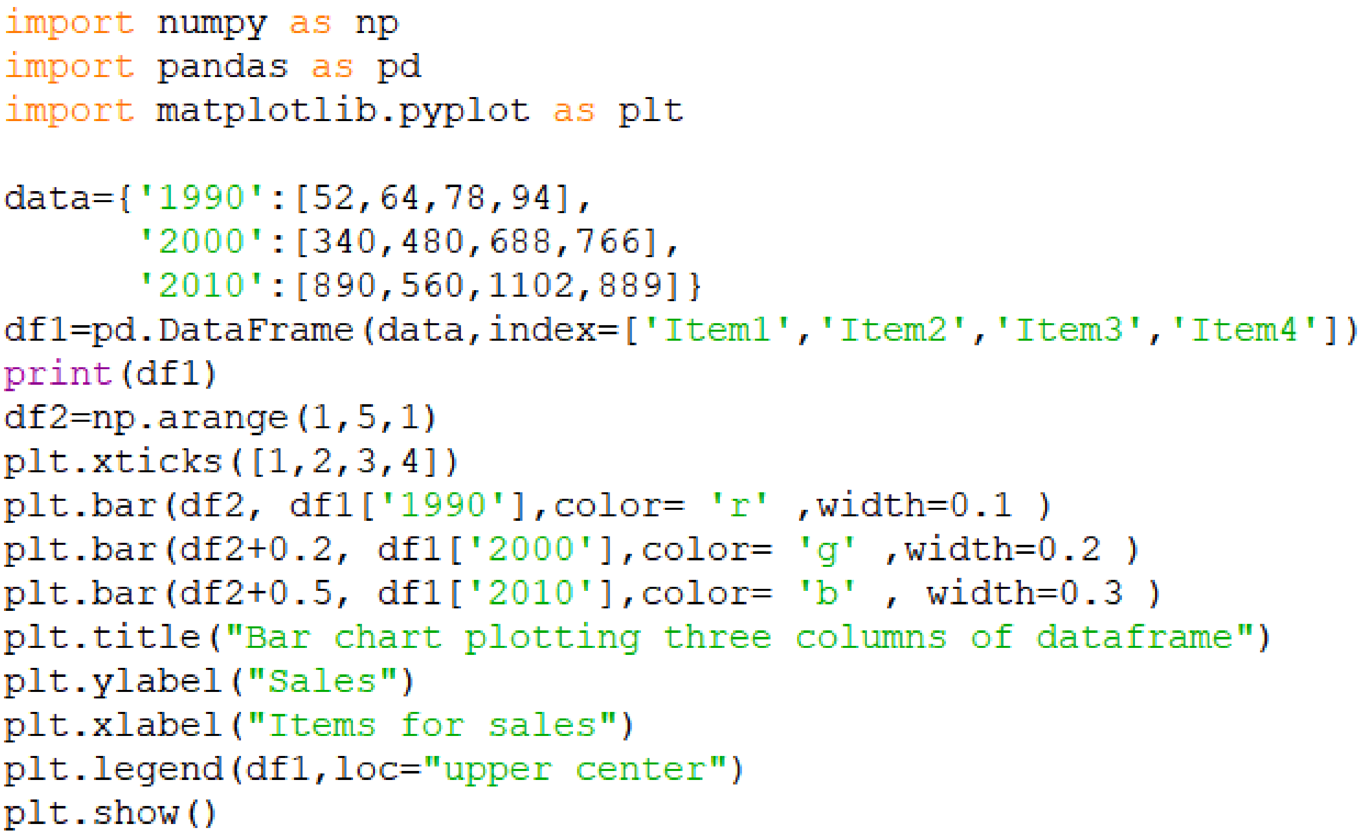


Figure 1
Analysis of the Performance of the students-subject wise
Abishek
Swaraj
Hitesh
Gokul
Omkar
Marks Obtained
0
10
20
30
40
50
60
70
80
IP
Eng
Eco
Bst
Acct
Subjects
Start

Practical 19-Write a program to create a bar chart plotting from the columns of DataFrame.

Solution:



```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt

data={'1990':[52,64,78,94],
      '2000':[340,480,688,766],
      '2010':[890,560,1102,889]}
df1=pd.DataFrame(data,index=['Item1','Item2','Item3','Item4'])
print(df1)
df2=np.arange(1,5,1)
plt.xticks([1,2,3,4])
plt.bar(df2, df1['1990'],color= 'r' ,width=0.1 )
plt.bar(df2+0.2, df1['2000'],color= 'g' ,width=0.2 )
plt.bar(df2+0.5, df1['2010'],color= 'b' , width=0.3 )
plt.title("Bar chart plotting three columns of dataframe")
plt.ylabel("Sales")
plt.xlabel("Items for sales")
plt.legend(df1,loc="upper center")
plt.show()
```

Output:

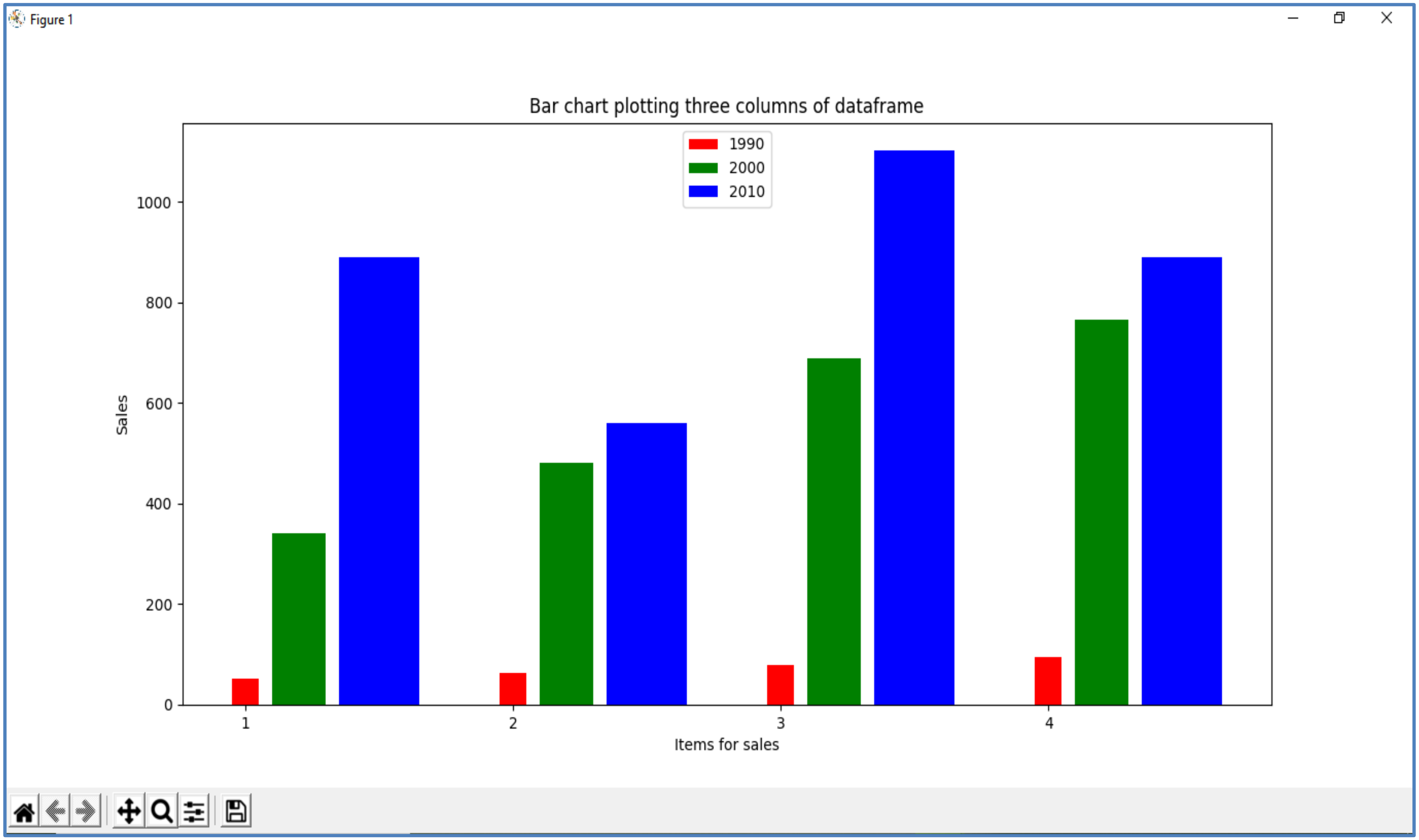
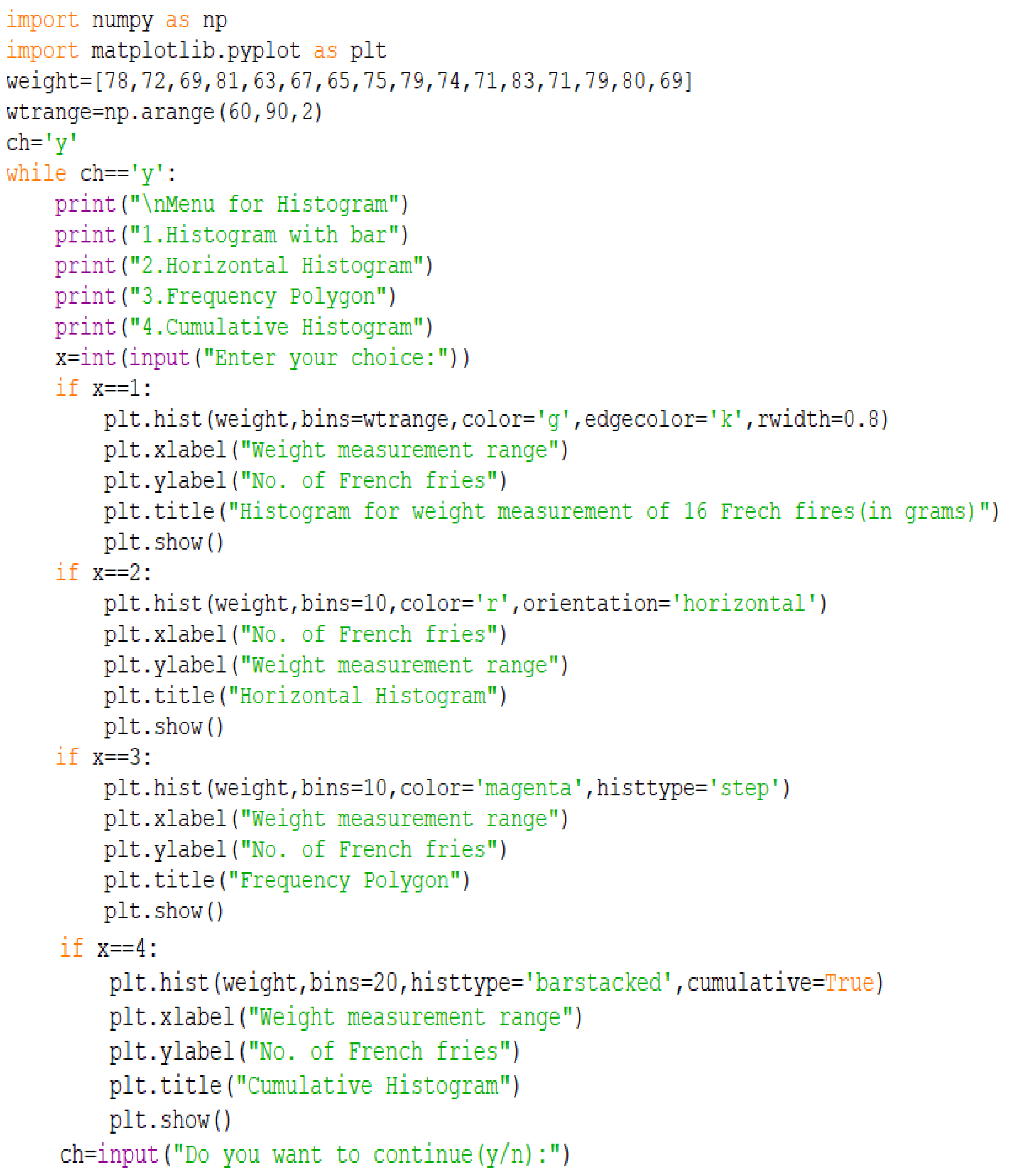


Figure 1
Bar chart plotting three columns of dataframe
🟥 1990
🟩 2000
🟦 2010
Sales
0
200
400
600
800
1000
1
2
3
4
Items for sales

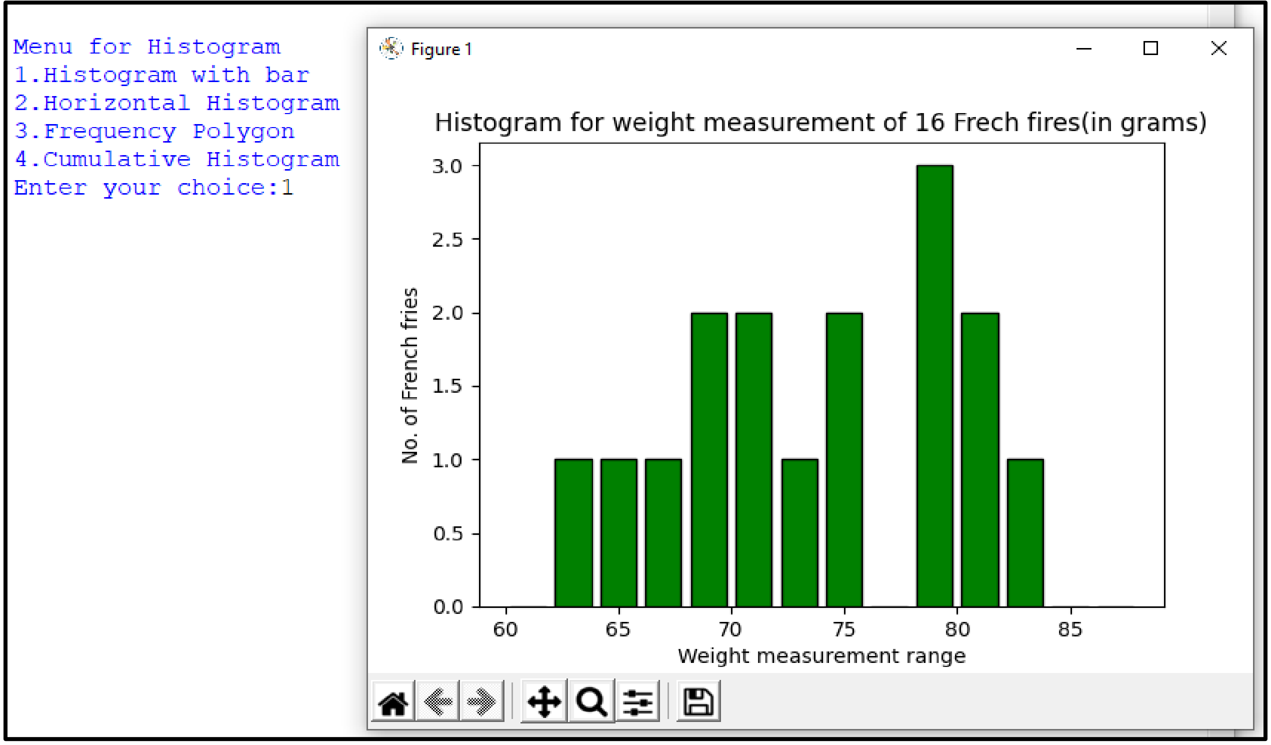
Practical 20-Write a program to create different histograms for the weight measurements for 16 small orders of French fries (in grams).

Solution:

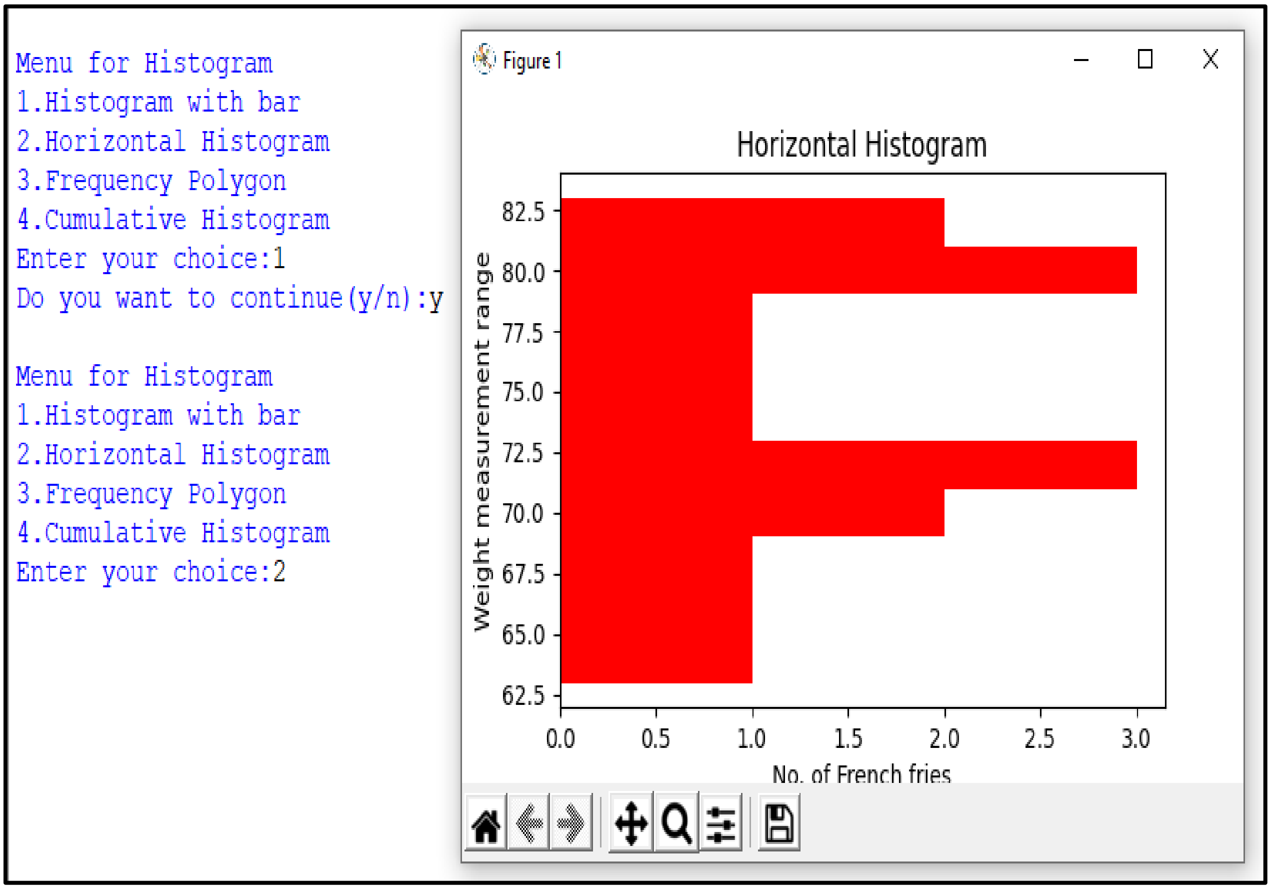


```
import numpy as np
import matplotlib.pyplot as plt
weight=[78,72,69,81,63,67,65,75,79,74,71,83,71,79,80,69]
wtrange=np.arange(60,90,2)
ch='y'
while ch=='y':
    print("\nMenu for Histogram")
    print("1.Histogram with bar")
    print("2.Horizontal Histogram")
    print("3.Frequency Polygon")
    print("4.Cumulative Histogram")
    x=int(input("Enter your choice:"))
    if x==1:
        plt.hist(weight,bins=wtrange,color='g',edgecolor='k',rwidth=0.8)
        plt.xlabel("Weight measurement range")
        plt.ylabel("No. of French fries")
        plt.title("Histogram for weight measurement of 16 Frech fires(in grams)")
        plt.show()
    if x==2:
        plt.hist(weight,bins=10,color='r',orientation='horizontal')
        plt.xlabel("No. of French fries")
        plt.ylabel("Weight measurement range")
        plt.title("Horizontal Histogram")
        plt.show()
    if x==3:
        plt.hist(weight,bins=10,color='magenta',histtype='step')
        plt.xlabel("Weight measurement range")
        plt.ylabel("No. of French fries")
        plt.title("Frequency Polygon")
        plt.show()
    if x==4:
        plt.hist(weight,bins=20,histtype='barstacked',cumulative=True)
        plt.xlabel("Weight measurement range")
        plt.ylabel("No. of French fries")
        plt.title("Cumulative Histogram")
        plt.show()
    ch=input("Do you want to continue(y/n):")
```

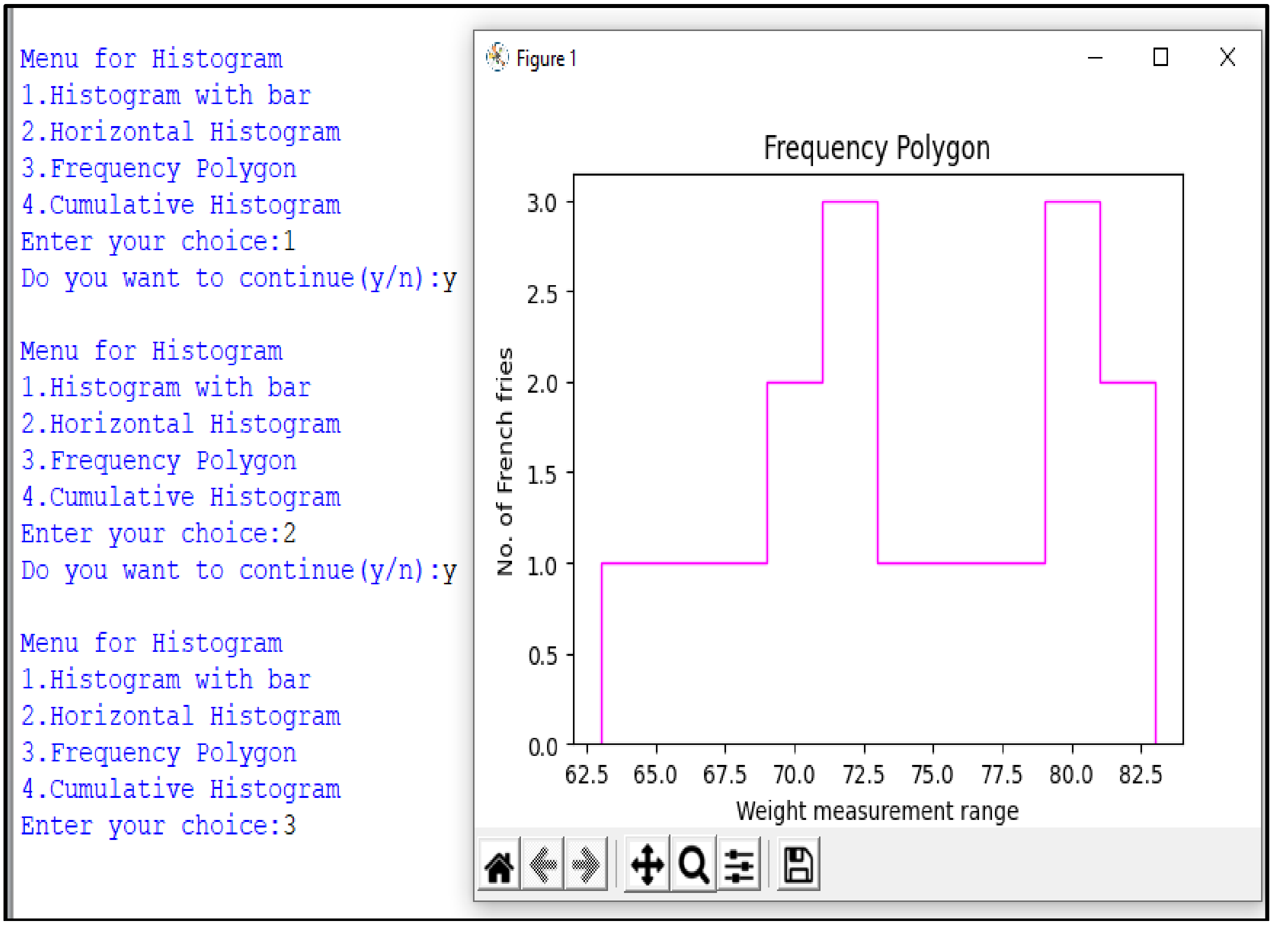
Output:



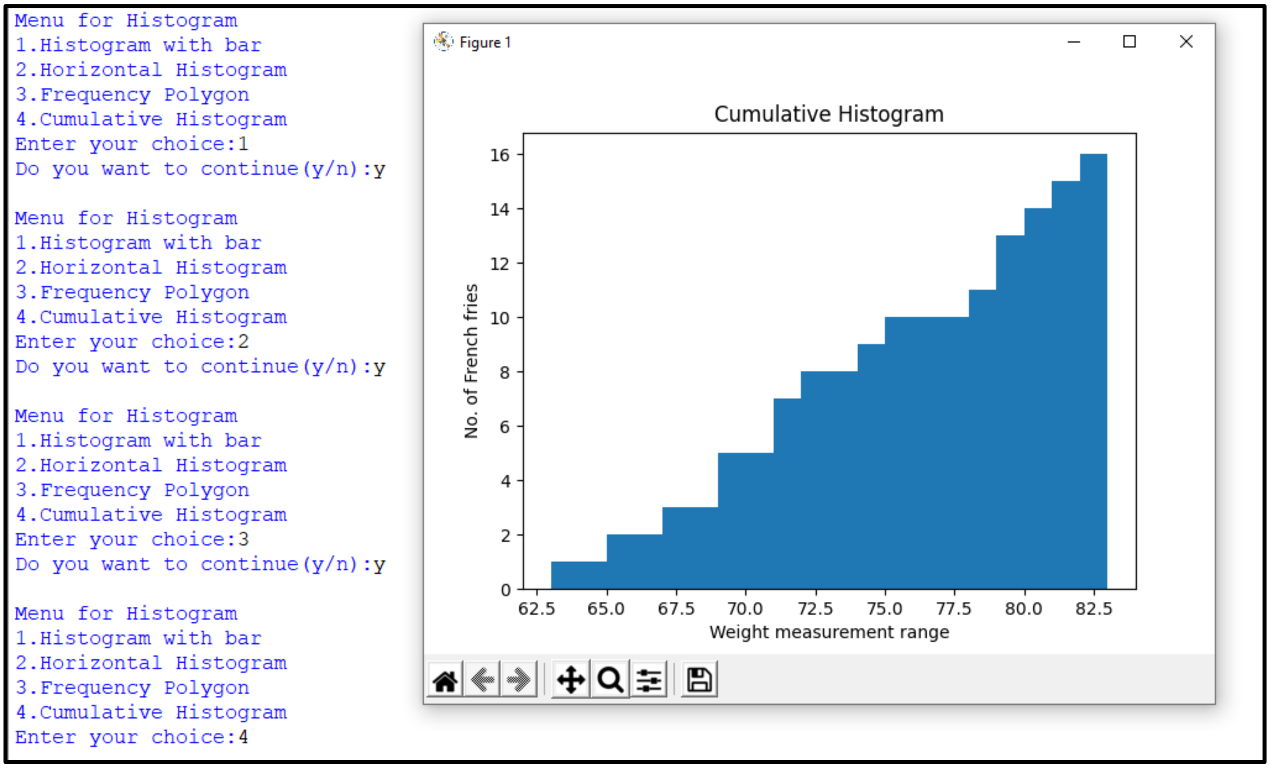
```
Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:1
```



```
Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:1
Do you want to continue(y/n):y

Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:2
```



```
Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:1
Do you want to continue(y/n):y

Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:2
Do you want to continue(y/n):y

Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:3
```



```
Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:1
Do you want to continue(y/n):y

Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:2
Do you want to continue(y/n):y

Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:3
Do you want to continue(y/n):y

Menu for Histogram
1.Histogram with bar
2.Horizontal Histogram
3.Frequency Polygon
4.Cumulative Histogram
Enter your choice:4
```