School Name

School Logo

PRACTICAL FILE

Informatics Practices / Computer Science (Python)

Name :

Class:

School:

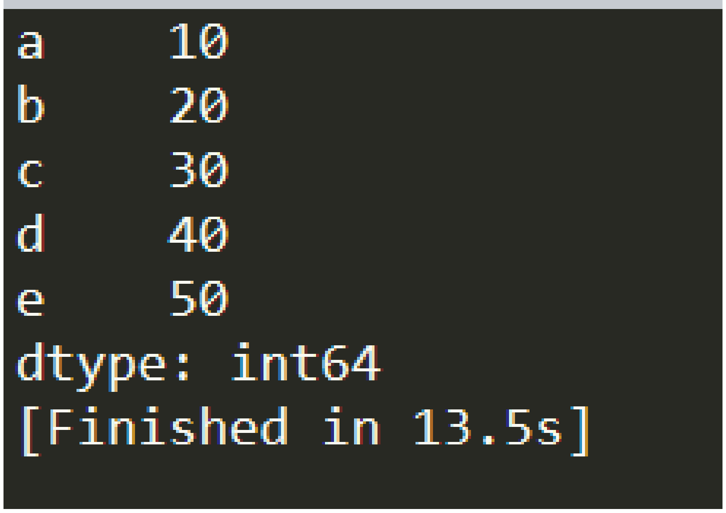
Roll No.

INDEX

1. Write a program to create a series using List with defined index values.

```
import pandas as pd
list1 = [10,20,30,40,50]
series1 = pd.Series(list1, index = ['a','b','c','d','e'])
print(series1)
```

Output

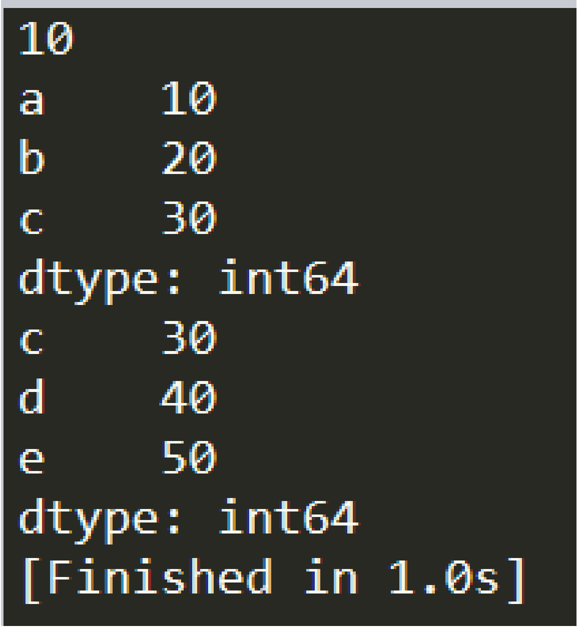


```
a    10
b    20
c    30
d    40
e    50
dtype: int64
[Finished in 13.5s]
```

2. Write a program to create a series using List with defined index values and slicing data from a series.

```
import pandas as pd
list1 = [10,20,30,40,50]
s = pd.Series(list1, index = ['a','b','c','d','e'])
print(s[0])         #for 0 index position
print(s[:3])#for first 3 index values
print(s[-3:])       #for last 3 index values
```

Output

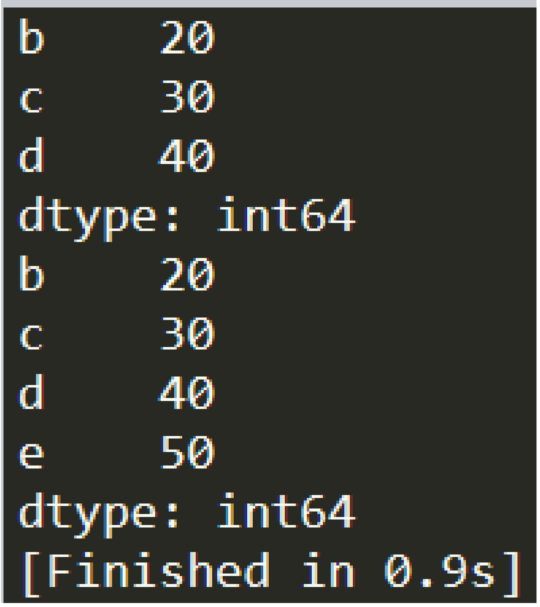


```
10
a    10
b    20
c    30
dtype: int64
c    30
d    40
e    50
dtype: int64
[Finished in 1.0s]
```

3. Write a program to create a series using List with defined index values and display the data using index wise and location wise.

```
import pandas as pd
list1 = [10,20,30,40,50]
s = pd.Series(list1, index = ['a','b','c','d','e'])
print(s.iloc[1:4])
print(s.loc['b':'e'])
```

Output

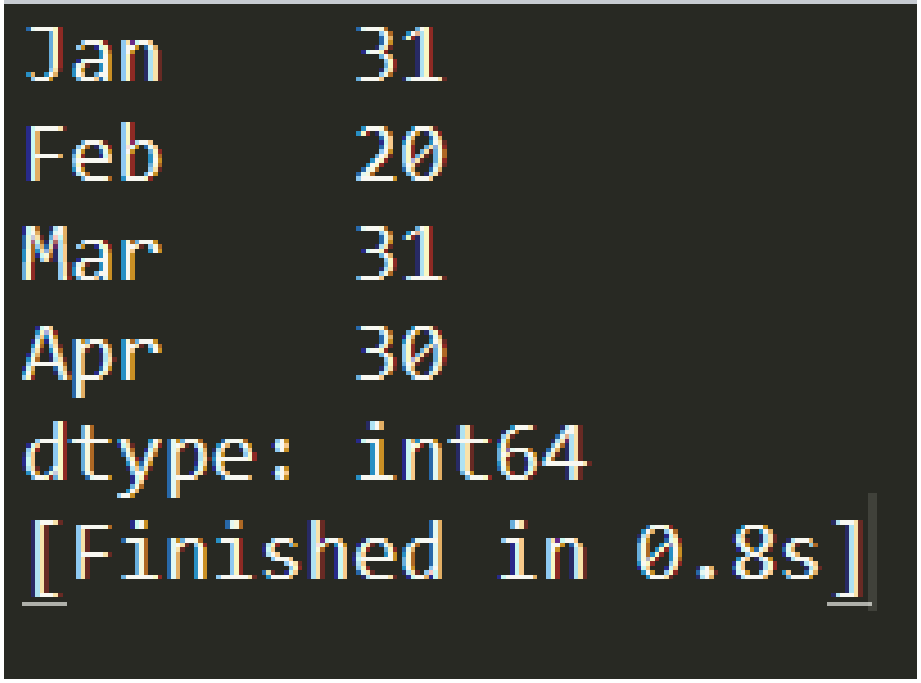


```
b    20
c    30
d    40
dtype: int64
b    20
c    30
d    40
e    50
dtype: int64
[Finished in 0.9s]
```

4. Write a program to create a series using dictionary

```
import pandas as pd
series = pd.Series({'Jan':31,'Feb':20,'Mar':31,'Apr':30})
print(series)
```

Output

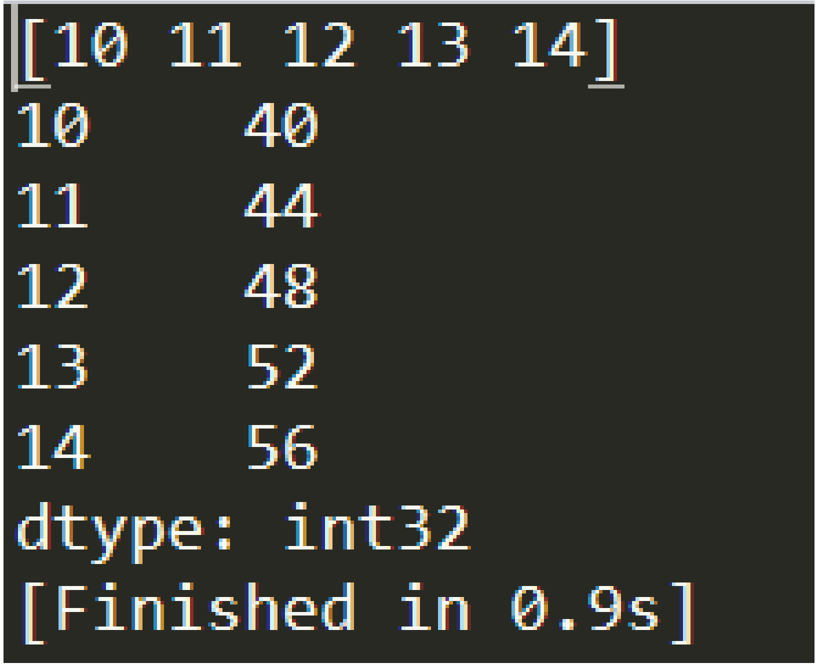


```
Jan    31
Feb    20
Mar    31
Apr    30
dtype: int64
[Finished in 0.8s]
```

5. Write a program to create a series through a mathematical expression

```
import pandas as pd
import numpy as np
s1 = np.arange(10,15)
print(s1)
subj=pd.Series(index = s1,data=s1*4)
print(subj)
```

Output

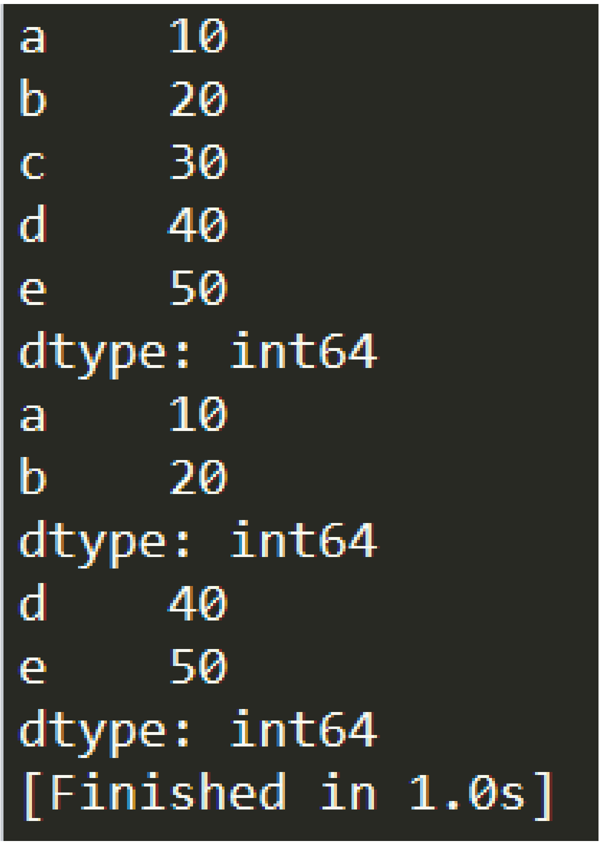


```
[10 11 12 13 14]
10    40
11    44
12    48
13    52
14    56
dtype: int32
[Finished in 0.9s]
```

6. Write a program to create a series using head() and tail() functions

```
import pandas as pd
series = pd.Series([10,20,30,40,50], index = ['a','b','c','d','e'])
print(series)
print(series.head(2))
#if no argument is provided then by default 5 rows are displayed
print(series.tail(2))
```

Output

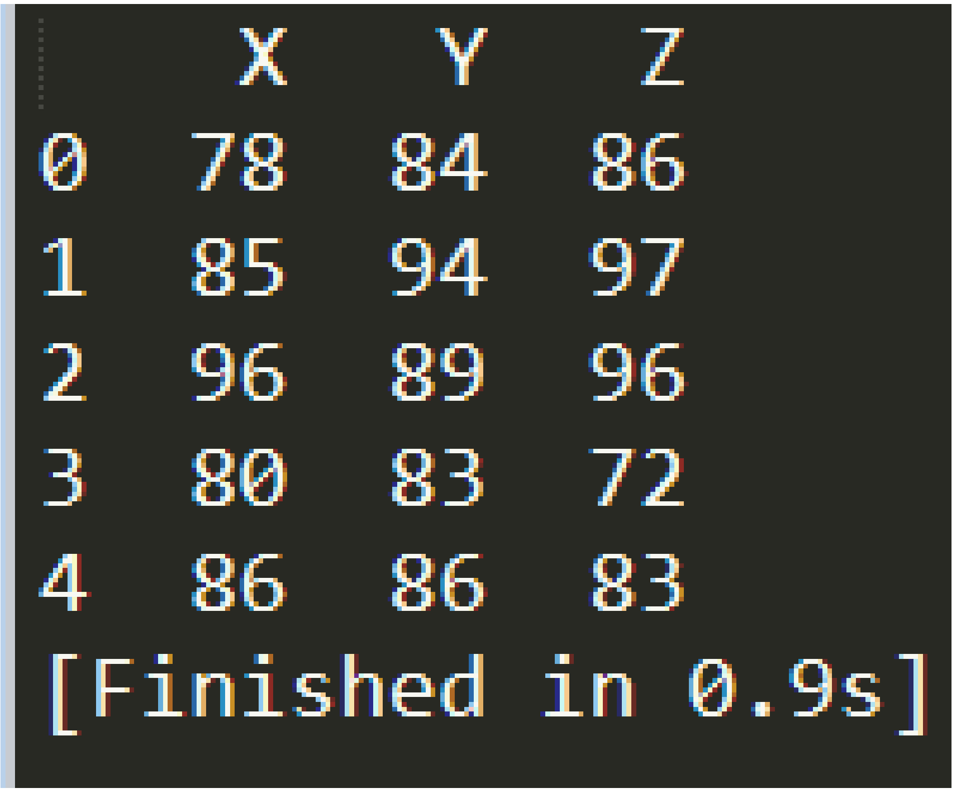


```
a    10
b    20
c    30
d    40
e    50
dtype: int64
a    10
b    20
dtype: int64
d    40
e    50
dtype: int64
[Finished in 1.0s]
```

7. Write a Pandas program to creat dataframe from a sample data Sample data: {'X':[78,85,96,80,86], 'Y':[84,94,89,83,86], 'Z':[86,97,96,72,83]}

```
import pandas as pd
df = pd.DataFrame({'X':[78,85,96,80,86], 'Y':[84,94,89,83,86],
        'Z':[86,97,96,72,83]});
print(df)
```

Output

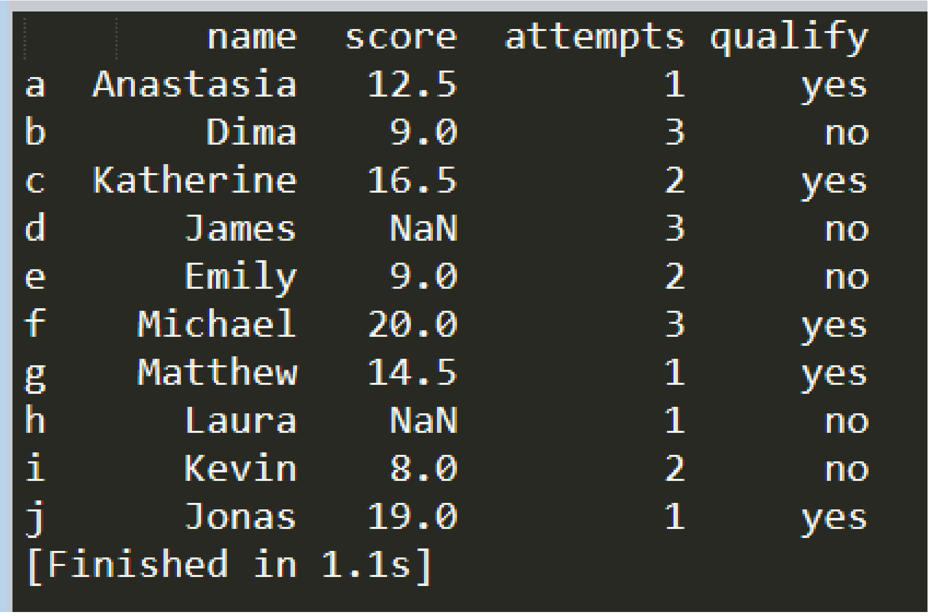


```
    X   Y   Z
0  78  84  86
1  85  94  97
2  96  89  96
3  80  83  72
4  86  86  83
[Finished in 0.9s]
```

8. Write a Pandas program to creat dataframe from a sample data

```
import pandas as pd
import numpy as np


exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James',
'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']


df = pd.DataFrame(exam_data , index=labels)
print(df)
```

Output

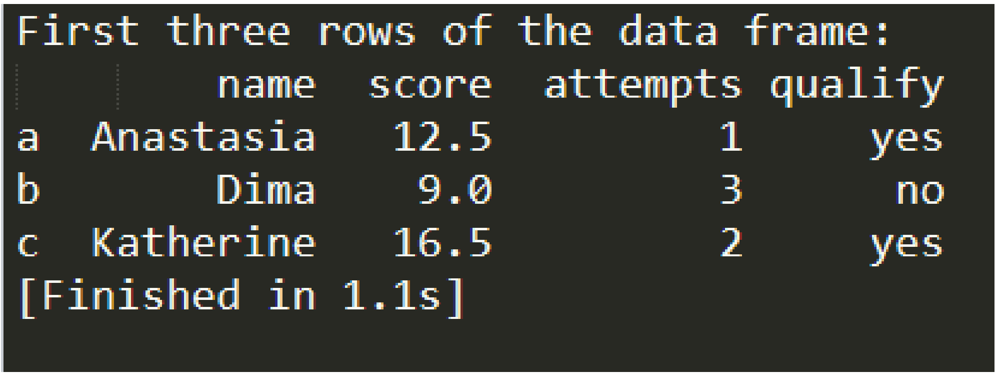


```
        name  score  attempts qualify
a  Anastasia   12.5         1     yes
b       Dima    9.0         3      no
c  Katherine   16.5         2     yes
d      James    NaN         3      no
e      Emily    9.0         2      no
f    Michael   20.0         3     yes
g    Matthew   14.5         1     yes
h      Laura    NaN         1      no
i      Kevin    8.0         2      no
j      Jonas   19.0         1     yes
[Finished in 1.1s]
```

9. Write a Pandas program to creat dataframe from a sample data

```
import pandas as pd
import numpy as np

exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine',
                    'James', 'Emily', 'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],
      'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],
      'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],
      'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no', 'no', 'yes']}
labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)
print("First three rows of the data frame:")
print(df.iloc[:3])
```

Output

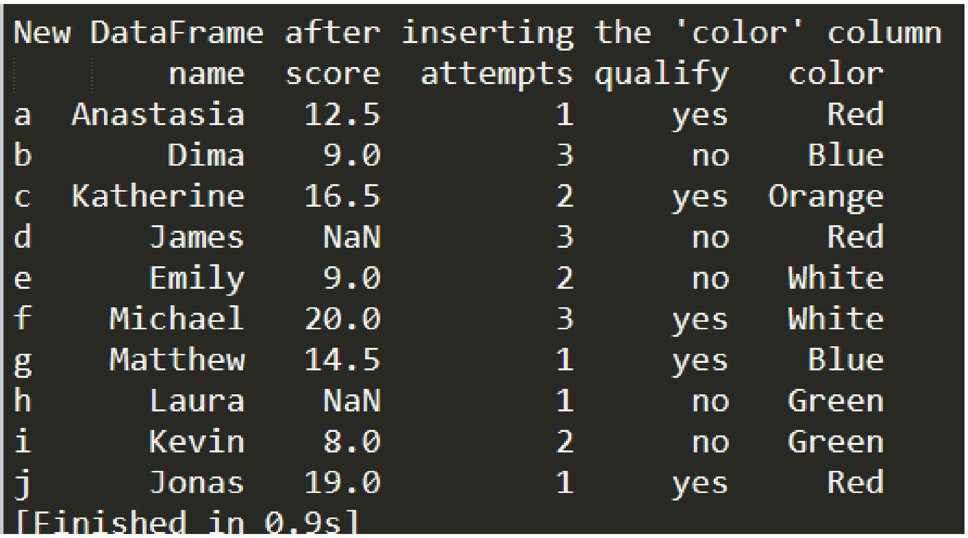


```
First three rows of the data frame:
        name  score  attempts qualify
a  Anastasia   12.5         1     yes
b       Dima    9.0         3      no
c  Katherine   16.5         2     yes
[Finished in 1.1s]
```

10. Write a Pandas program to creat dataframe from a sample data and create one column with different name of colours

```
import pandas as pd

import numpy as np

exam_data  = {'name': ['Anastasia', 'Dima', 'Katherine', 'James', 'Emily',

                        'Michael', 'Matthew', 'Laura', 'Kevin', 'Jonas'],

        'score': [12.5, 9, 16.5, np.nan, 9, 20, 14.5, np.nan, 8, 19],

        'attempts': [1, 3, 2, 3, 2, 3, 1, 1, 2, 1],

        'qualify': ['yes', 'no', 'yes', 'no', 'no', 'yes', 'yes', 'no',

        'no', 'yes']}

labels = ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i', 'j']

df = pd.DataFrame(exam_data , index=labels)

print("Original rows:")

print(df)

color =

['Red','Blue','Orange','Red','White','White','Blue','Green','Green','Red']

df['color'] = color

print("\nNew DataFrame after inserting the 'color' column")

print(df)
```

Output

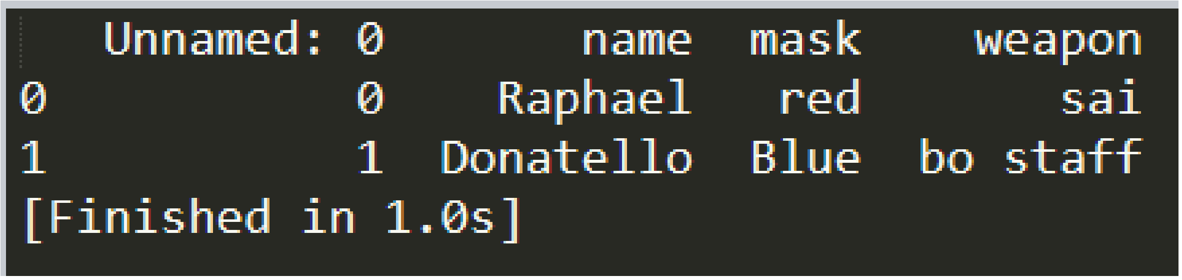


```
New DataFrame after inserting the 'color' column
        name  score  attempts qualify   color
a  Anastasia   12.5         1     yes     Red
b       Dima    9.0         3      no    Blue
c  Katherine   16.5         2     yes  Orange
d      James    NaN         3      no     Red
e      Emily    9.0         2      no   White
f    Michael   20.0         3     yes   White
g    Matthew   14.5         1     yes    Blue
h      Laura    NaN         1      no   Green
i      Kevin    8.0         2      no   Green
j      Jonas   19.0         1     yes     Red
[Finished in 0.9s]
```

11. Write a program to read csv file using pandas

```
import pandas as pd

data = pd.read_csv("file2.csv")

print(data.head())
```

Output

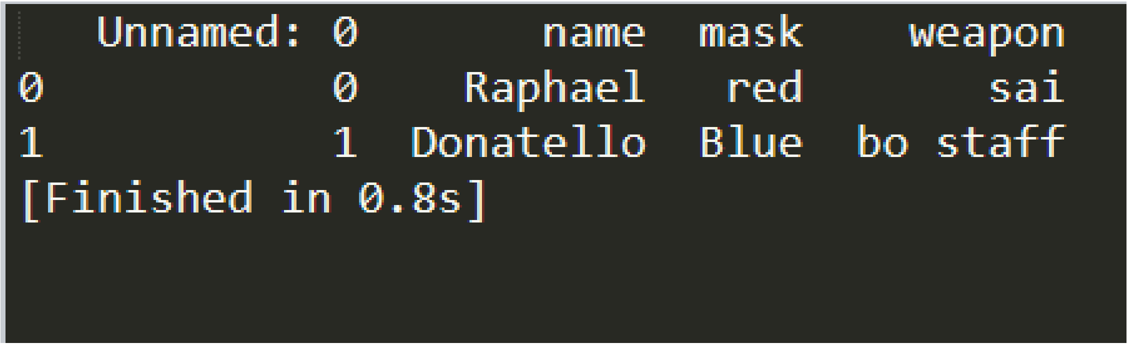


```
   Unnamed: 0        name  mask    weapon
0           0     Raphael   red       sai
1           1  Donatello  Blue  bo staff
[Finished in 1.0s]
```

12. Write a program to write the data in file1.csv file using pandas

```
import pandas as pd

df = pd.DataFrame({'name': ['Raphael', 'Donatello'],
                   'mask': ['red', 'Blue'],
                   'weapon': ['sai', 'bo staff']})
df.to_csv("file2.csv")
```

Output

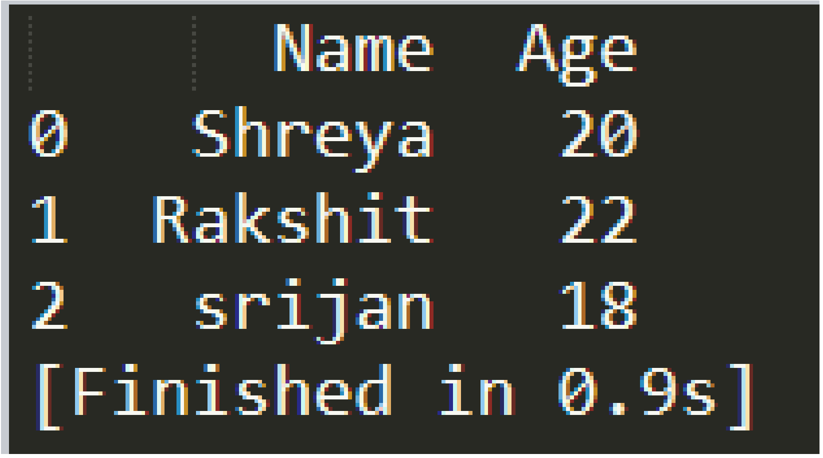


```
   Unnamed: 0       name  mask    weapon
0           0    Raphael   red       sai
1           1  Donatello  Blue  bo staff
[Finished in 0.8s]
```

13. Write a program to creat DataFrame from Lists with column heading

```
import pandas as pd
data1 = [['Shreya', 20],['Rakshit', 22], ['srijan',18]]
df1 = pd.DataFrame(data1, columns=['Name','Age']) #Defining column names to be displayed as headings
print(df1)
```

Output

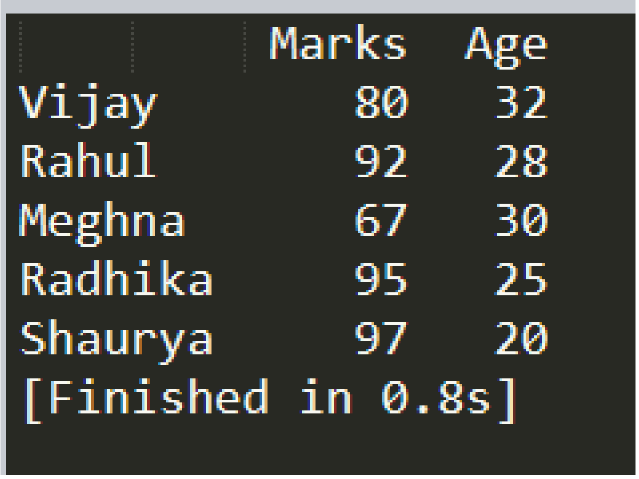


```
       Name  Age
0    Shreya   20
1  Rakshit   22
2    srijan   18
[Finished in 0.9s]
```

14. write a program to create dataframe from two series of student data

```
import pandas as pd

student_marks = pd.Series({'Vijay':80,'Rahul':92,'Meghna':67,'Radhika':95,'Shaurya':97})

student_age = pd.Series({'Vijay':32,'Rahul':28,'Meghna':30,'Radhika':25,'Shaurya':20})

student_df = pd.DataFrame({'Marks':student_marks,'Age':student_age})

print(student_df)
```

Output

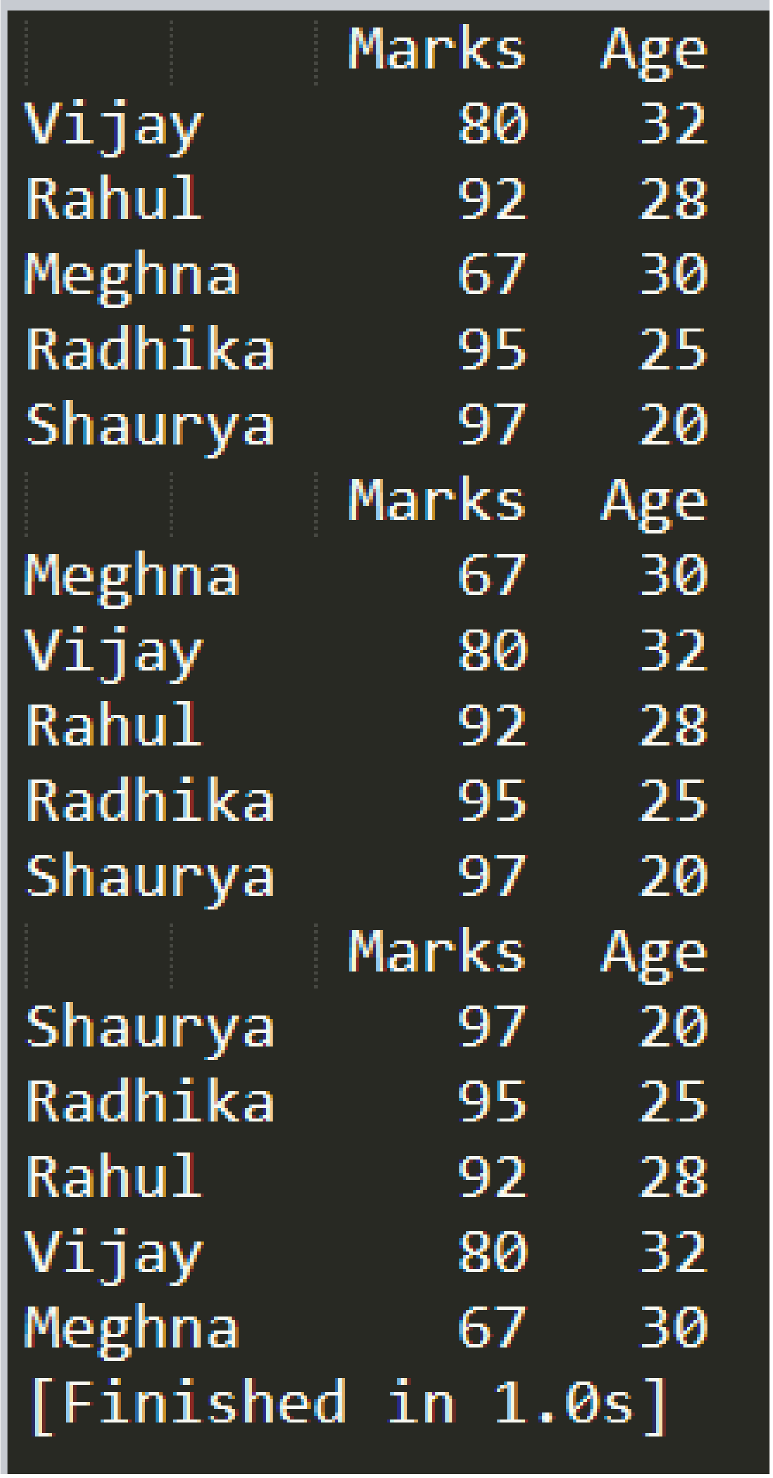


```
         Marks  Age
Vijay       80   32
Rahul       92   28
Meghna      67   30
Radhika     95   25
Shaurya     97   20
[Finished in 0.8s]
```

15. write a program to create dataframe from two series of student and sort data in ascending and descending order by using marks

```
import pandas as pd

student_marks = pd.Series({'Vijay':80,'Rahul':92,'Meghna':67,'Radhika':95,'Shaurya':97})

student_age = pd.Series({'Vijay':32,'Rahul':28,'Meghna':30,'Radhika':25,'Shaurya':20})

student_df = pd.DataFrame({'Marks':student_marks,'Age':student_age})

print(student_df)

#sorting the data on the basis of marks in ascending order

print(student_df.sort_values(by=['Marks'])) #by keyword defines

#the field on the basis of which the data is to be sorted

print(student_df.sort_values(by=['Marks'],ascending=False))

#sorted in descending order of Marks
```

Output



```
         Marks  Age
Vijay       80   32
Rahul       92   28
Meghna      67   30
Radhika     95   25
Shaurya     97   20
         Marks  Age
Meghna      67   30
Vijay       80   32
Rahul       92   28
Radhika     95   25
Shaurya     97   20
         Marks  Age
Shaurya     97   20
Radhika     95   25
Rahul       92   28
Vijay       80   32
Meghna      67   30
[Finished in 1.0s]
```

16. write a program to plot the elements of two lists using a bar chart.

```
import matplotlib.pyplot as plt

x = [2,4,6,8,10]
y = [6,7,8,2,4]

x2 = [1,3,5,7,9]
y2 = [7,8,2,4,2]

plt.bar(x,y, label="Bars1")
plt.bar(x2,y2, label="Bars2")

plt.xlabel('x')
plt.ylabel('y')

plt.title('Bar Graph \n with multiline title')
plt.legend()
plt.show()
```

Output

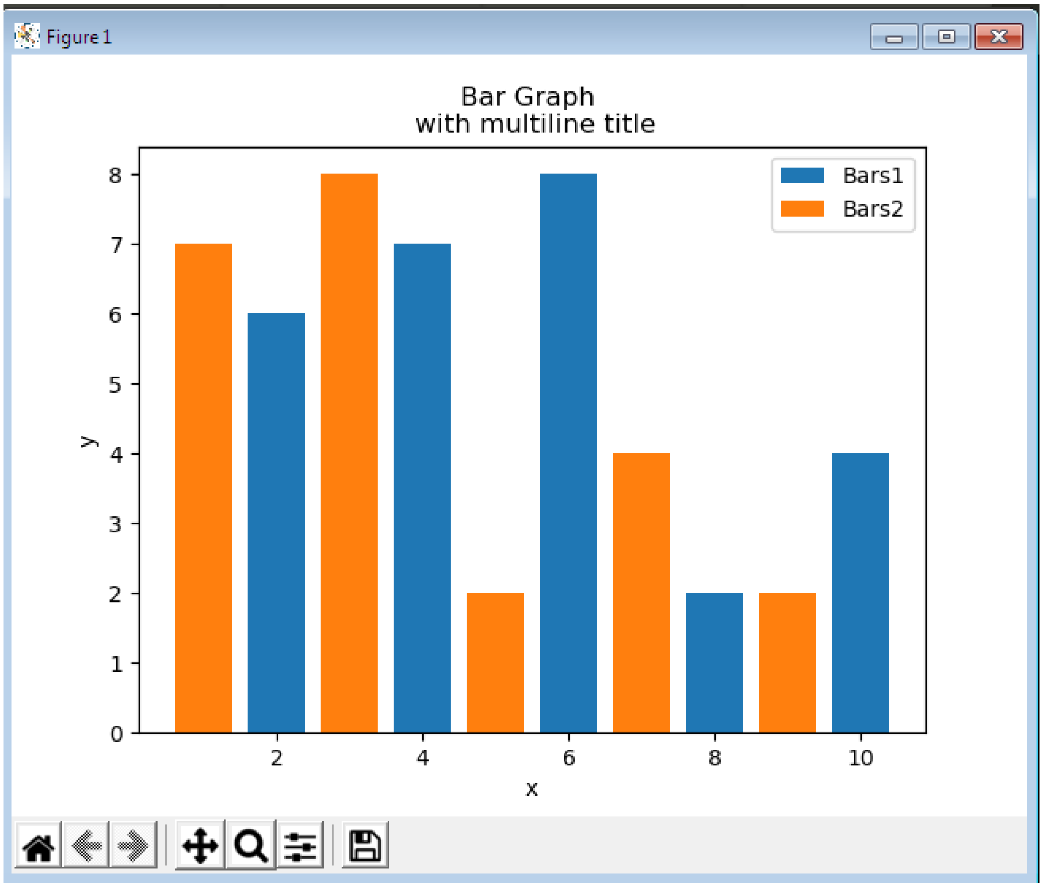
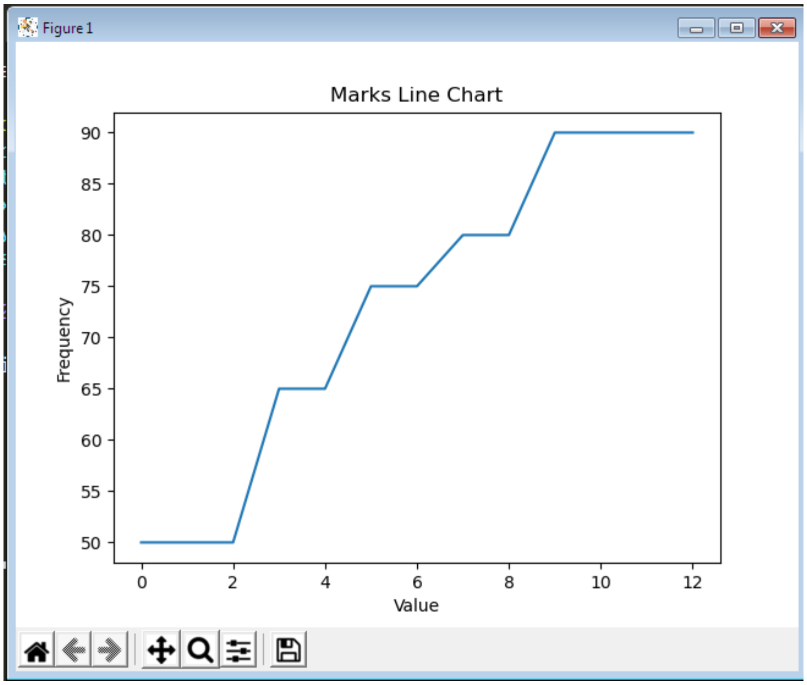


Figure 1
Bar Graph
with multiline title
y
x
0
1
2
3
4
5
6
7
8
2
4
6
8
10
🟦 Bars1
🟧 Bars2

17. write a program to plot frequency of marks using Line chart

```
import matplotlib.pyplot as plt

def fnplot(list1):
    plt.plot(list1)
    plt.title("Marks Line Chart")
    plt.xlabel("Value")
    plt.ylabel("Frequency")
    plt.show()

list1=[50,50,50,65,65,75,75,80,80,90,90,90,90]

fnplot(list1)
```

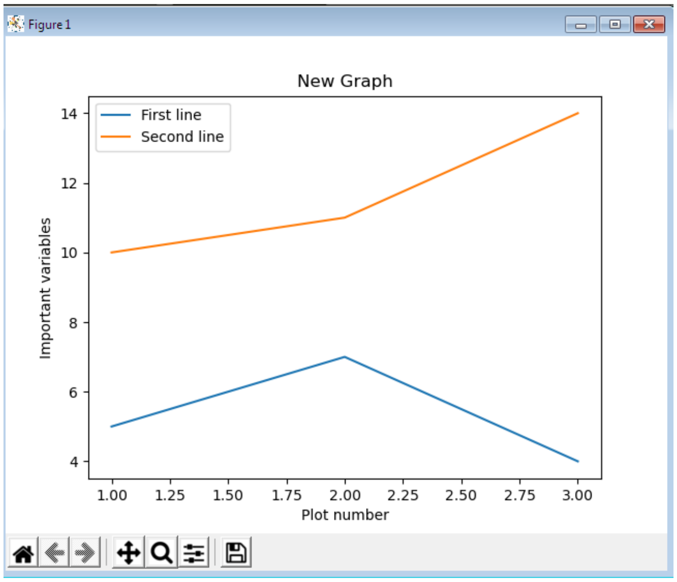
Output



18. write a program to draw two lines along with proper titles and legends

```
import matplotlib.pyplot as plt

x = [1,2,3]
y = [5,7,4]
plt.plot(x,y,label='First line')

x2 = [1,2,3]
y2 = [10,11,14]
plt.plot(x2,y2,label='Second line')
plt.xlabel('Plot number')
plt.ylabel('Important variables')
plt.title('New Graph')
plt.legend()

plt.show()
```

Output



19. write a program to depict the relationshi between unemployment Rate and Stock index price through a scatter plot.

Unemployment_Rate	Stock_Index_Price
6.1	1500
5.8	1520
5.7	1525
5.7	1523
5.8	1515
5.6	1540
5.5	1545
5.3	1560
5.2	1555
5.2	1565

```
import matplotlib.pyplot as plt


Unemployment_Rate = [6.1,5.8,5.7,5.7,5.8,5.6,5.5,5.3,5.2,5.2]
Stock_Index_Price =
[1500,1520,1525,1523,1515,1540,1545,1560,1555,1565]


plt.scatter(Unemployment_Rate, Stock_Index_Price, color='green')
plt.title('Unemployment_Rate Vs Stock_Index_Price', fontsize=14)
plt.xlabel('Unemployment_Rate', fontsize=14)
plt.ylabel('Stock_Index_Price', fontsize=14)
plt.grid(True)
plt.show()
```

Output

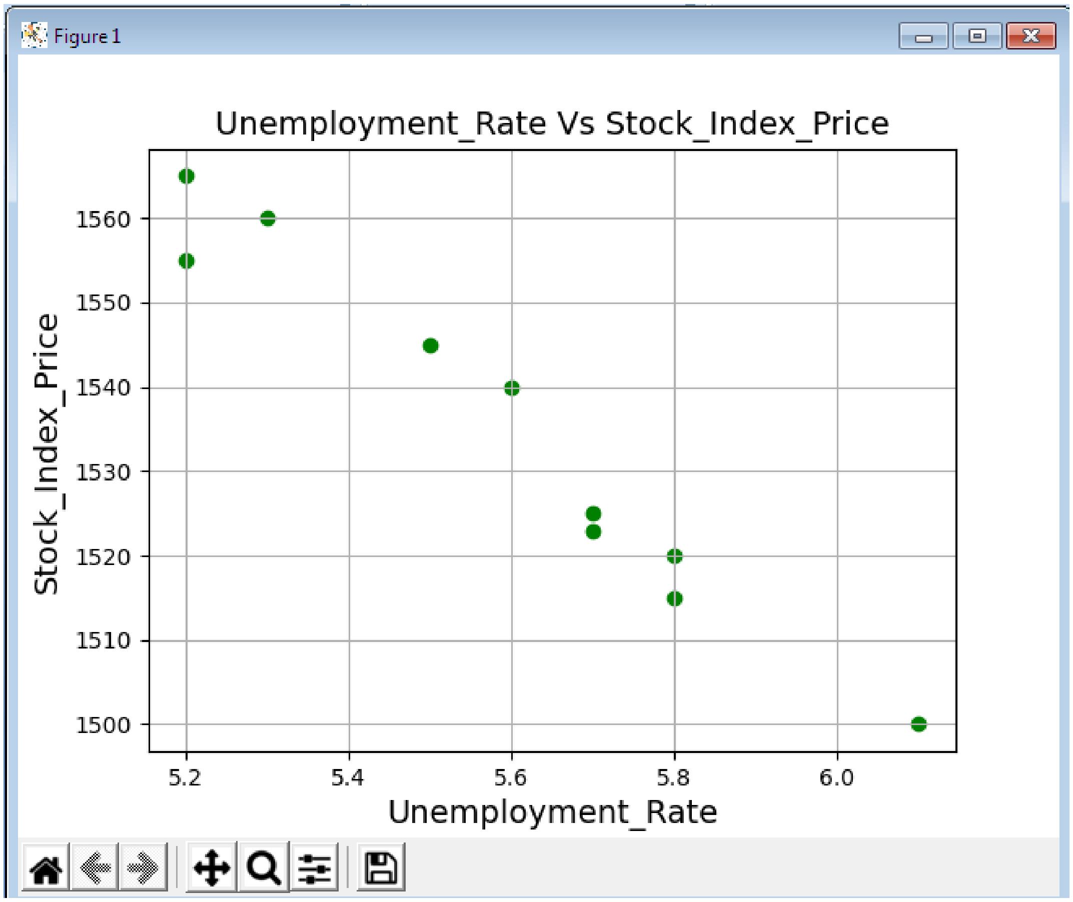


Figure 1
Unemployment_Rate Vs Stock_Index_Price
Stock_Index_Price
1560
1550
1540
1530
1520
1510
1500
5.2
5.4
5.6
5.8
6.0
Unemployment_Rate

20. Write a Python program to plot two or more lines with different styles (dotted lines)

```
import matplotlib.pyplot as plt

x1 = [10,20,30]
y1 = [20,40,10]

x2 = [10,20,30]
y2 = [40,10,30]

plt.xlabel('X - axis')
plt.ylabel('Y - axis')

plt.plot(x1,y1, color='blue', linewidth=3, label='Line1-dotted',linestyle='dotted')

plt.plot(x2,y2, color='red', linewidth=5, label='Line2-deshed',linestyle='dashed')

plt.title("Plot with two or more lines with different styles")

plt.legend()

plt.show()
```

Output

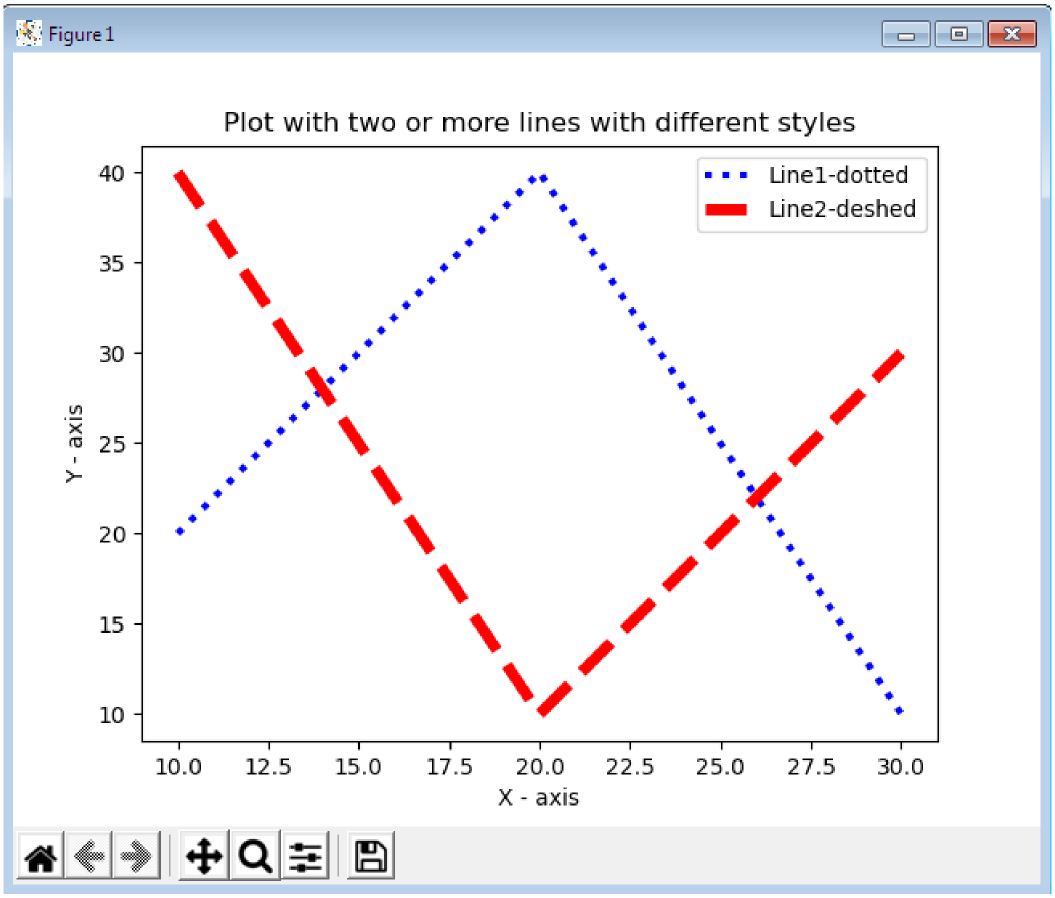
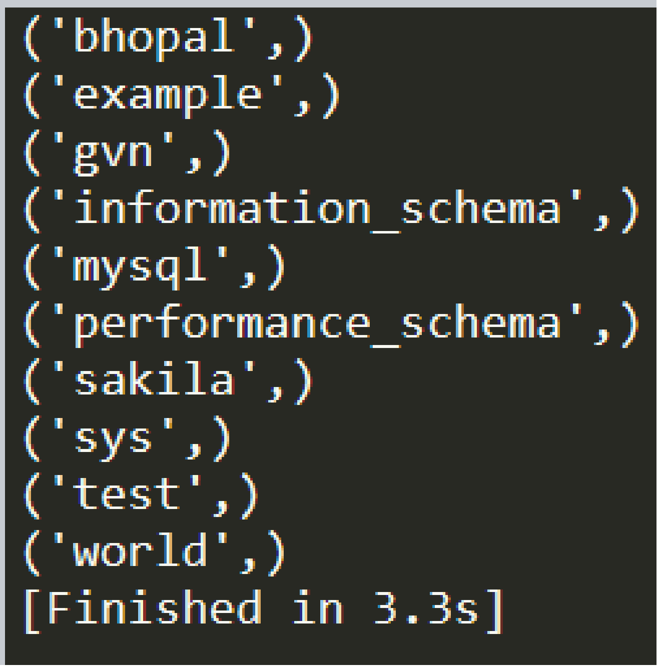


Figure 1
Plot with two or more lines with different styles
Line1-dotted
Line2-deshed
Y - axis
X - axis
40
35
30
25
20
15
10
10.0
12.5
15.0
17.5
20.0
22.5
25.0
27.5
30.0

21. Write a Python program to established connection with MySQL and display all the data

```
import mysql.connector
mydb = mysql.connector.connect(host="localhost",user="root",passwd="1234")

mycursor = mydb.cursor()
mycursor.execute("show databases")

for x in mycursor:
   print(x)
```

Output

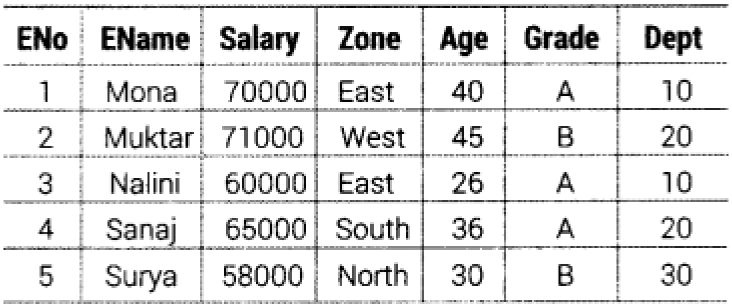


```
('bhopal',)
('example',)
('gvn',)
('information_schema',)
('mysql',)
('performance_schema',)
('sakila',)
('sys',)
('test',)
('world',)
[Finished in 3.3s]
```

22. Write a Python program to established connection with MySQL and insert some data into client table

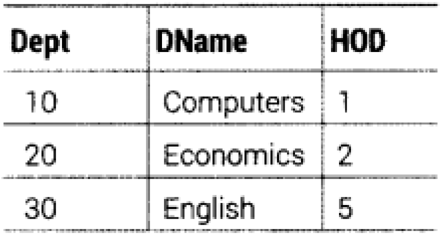
```
import mysql.connector
mydb = mysql.connector.connect(host="localhost",user="root",passwd="1234",database="gvn")



mycursor = mydb.cursor()
mycursor.execute("insert into client values(6,'mahesh','sagar',128)")

mydb.commit()

print(mycursor.rowcount, "record inserted.")
```

In a database BANK, there are two tables with a sample data given below
TABLE EMPLOYEE



ENo	EName	Salary	Zone	Age	Grade	Dept
1	Mona	70000	East	40	A	10
2	Muktar	71000	West	45	B	20
3	Nalini	60000	East	26	A	10
4	Sanaj	65000	South	36	A	20
5	Surya	58000	North	30	B	30

TABLE DEPARTMENT



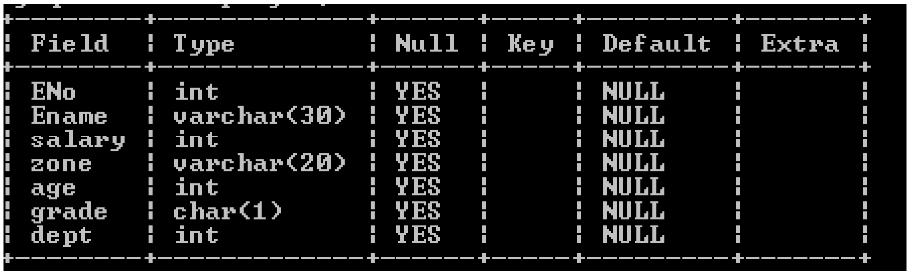
Dept	DName	HOD
10	Computers	1
20	Economics	2
30	English	5

Note:

– EName refers to Employee Name
– DName refers to Department Name
– Dept refers to Department Code
– HOD refers to Employee number (ENO) of the Head of the Department

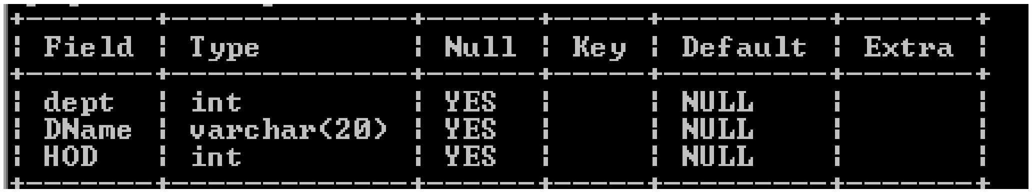
Write SQL queries for the following:

1. Create table EMPLOYEE as per following Table Instance Chart



```
+--------+-------------+------+-----+---------+-------+
¦ Field  ¦ Type        ¦ Null ¦ Key ¦ Default ¦ Extra ¦
+--------+-------------+------+-----+---------+-------+
¦ ENo    ¦ int         ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ Ename  ¦ varchar(30) ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ salary ¦ int         ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ zone   ¦ varchar(20) ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ age    ¦ int         ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ grade  ¦ char(1)     ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ dept   ¦ int         ¦ YES  ¦     ¦ NULL    ¦       ¦
+--------+-------------+------+-----+---------+-------+
```

2. Create table DEPARTMENT as per following Table Instance Chart



```
+-------+-------------+------+-----+---------+-------+
¦ Field ¦ Type        ¦ Null ¦ Key ¦ Default ¦ Extra ¦
+-------+-------------+------+-----+---------+-------+
¦ dept  ¦ int         ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ DName ¦ varchar(20) ¦ YES  ¦     ¦ NULL    ¦       ¦
¦ HOD   ¦ int         ¦ YES  ¦     ¦ NULL    ¦       ¦
+-------+-------------+------+-----+---------+-------+
```

3. To display EName, Zone, Salary of all the employees

4. To display ENo, EName, Salary and corresponding DName of all the employees whose age is between 25 and 35 (both values inclusive).

5. To display DName and corresponding EName from the tables DEPARTMENT and EMPLOYEE, (Hint’ HOD of the DEPARTMENT table should be matched with ENo of the EMPLOYEE table for getting the desired result).

Ans:

```
CREATE TABLE EMPLOYEE
(ENo int,
 EName VARCHAR(20),
 Salary int,
 Zone VARCHAR(10),
 Age int,
 Grade char,
 Dept int);
```

```
CREATE TABLE DEPARTMENT
(Dept  int,
 DName VARCHAR(30),
 HOD int);
```

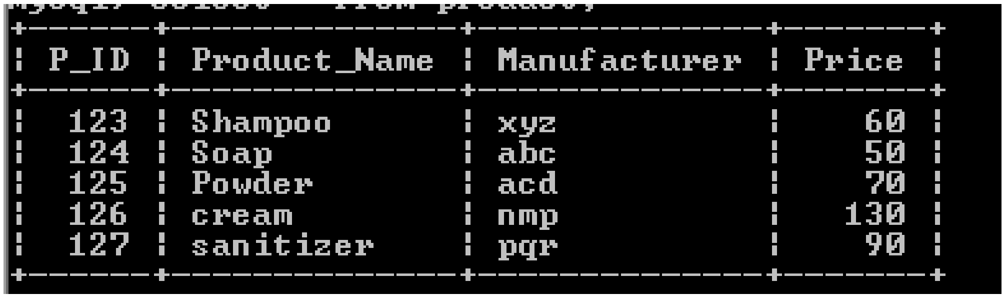
```
SELECT EName, Zone, Salary FROM EMPLOYEE;
```

```
SELECT e.ENo,e.EName,e.Salary,d.DName
FROM EMPLOYEE e, DEPARTMENT d
WHERE e.Dept = d.Dept AND e.Age BETWEEN 25 AND 35;
```

```
SELECT e.EName,d.DName
FROM EMPLOYEE e, DEPARTMENT d
WHERE e.ENO = d.HOD;
```

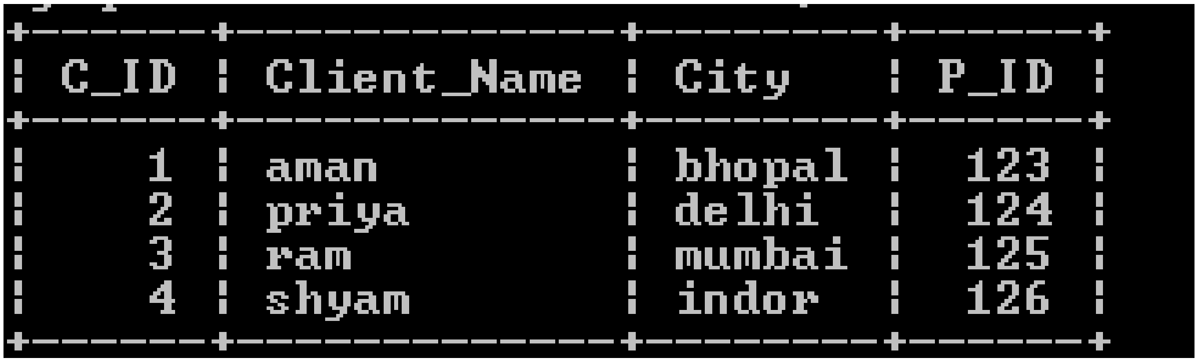
5.b) Consider the following tables Product and Client. Write SQL commands for the statement (i) to (iv) and give outputs for SQL queries (v) to (vi)

Table: **PRODUCT**



```
mysql> select * from product;
+------+--------------+--------------+-------+
| P_ID | Product_Name | Manufacturer | Price |
+------+--------------+--------------+-------+
|  123 | Shampoo      | xyz          |    60 |
|  124 | Soap         | abc          |    50 |
|  125 | Powder       | acd          |    70 |
|  126 | cream        | nmp          |   130 |
|  127 | sanitizer    | pqr          |    90 |
+------+--------------+--------------+-------+
```

Table: CLIENT



```
+------+-------------+--------+------+
| C_ID | Client_Name | City   | P_ID |
+------+-------------+--------+------+
|    1 | aman        | bhopal |  123 |
|    2 | priya       | delhi  |  124 |
|    3 | ram         | mumbai |  125 |
|    4 | shyam       | indor  |  126 |
+------+-------------+--------+------+
```

(i) To display the details of those Clients whose city is Delhi.

Ans: select * from Client where city='Delhi';

(ii) To display the details of Products whose Price is in the range of 50 to 100(Both values included).

Ans: select * from product where price between 50 and 100;

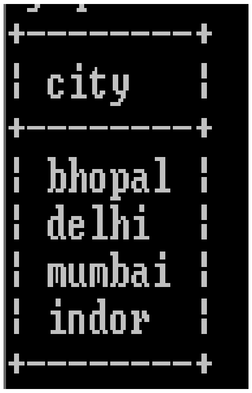
(iii) To display the ClientName, City from table Client, and ProductName and Price from table Product, with their corresponding matching P_ID.

```
Select c.client_name,c.city,p.product_name,p.price from
client c, product p where c.p_id = p.p_id;
```

(iv) To increase the Price of all Products by 10

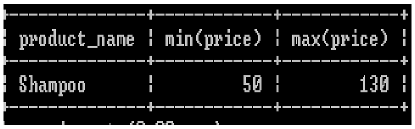
Ans: Update Product Set Price=Price+10;

(v) SELECT DISTINCT City FROM Client.



```
+--------+
| city   |
+--------+
| bhopal |
| delhi  |
| mumbai |
| indor  |
+--------+
```

(vi) select product_name, min(price), max(price) from product;



```
+--------------+------------+------------+
| product_name | min(price) | max(price) |
+--------------+------------+------------+
| Shampoo      |         50 |        130 |
+--------------+------------+------------+
```

TABLE: FLIGHTS

FL_NO	**STARTING**	**ENDING**	**NO_ FLGHTS**	**NO_ STOPS**
IC301	MUMBAI	DELHI	8	0
IC799	BANGALORE	DELHI	2	1
MC101	INDORE	MUMBAI	3	0
IC302	DELHI	MUMBAI	8	0
AM812	KANPUR	BANGLORE	3	1
IC899	MUMBAI	KOCHI	1	4
AM501	DELHI	TRIVENDRUM	1	5
MU499	MUMBAI	MADRAS	3	3
IC701	DELHI	AHMEDABAD	4	0

TABLE:FLIGHTS

FL_NO	**AIRLINES**	**FARE**	**TAX%**
IC701	INDIAN AIRLINES	6500	10
MU499	SAHARA	9400	5
AM501	JET AIRWAYS	13450	8
IC899	INDIAN AIRLINES	8300	4
IC302	INDIAN AIRLINES	4300	10
IC799	INDIAN AIRLINES	1050	10
MC101	DECCAN AIRLINES	3500	4

(i) Display FL_NO and NO_FLIGHTS from “KANPUR” TO “BANGALORE” from the table FLIGHTS.

Ans: Select FL_NO, NO_FLIGHTS from FLIGHTS where Starting=”KANPUR” AND ENDING=”BANGALORE”

(ii) Arrange the contents of the table FLIGHTS in the ascending order of FL_NO.

Ans: (Children, Try this as an assignment)

(iii) Display the FL_NO and fare to be paid for the flights from DELHI to MUMBAI using the tables FLIGHTS and FARES, where the fare to paid = FARE+FARE+TAX%/100.

Ans: Select FL_NO, FARE+FARE+(TAX%/100) from FLIGHTS, FARES where Starting=”DELHI” AND Ending=”MUMBAI”

(iv) Display the minimum fare “Indian Airlines” is offering from the tables FARES.

Ans: Select min(FARE) from FARES Where AIRLINES=”Indian Airlines”

v) Select FL_NO,NO_FLIGHTS,AIRLINES from FLIGHTS, FARES Where STARTING = “DELHI” AND FLIGHTS.FL_NO = FARES.FL_NO

Ans: FL_NO NO_FLIGHTS AIRLINES IC799 2 Indian Airlines