

PM SHRI KENDRIYA VIDYALAYA AFS MANAURI, PRAYAGRAJ
LIST OF PRATICALS
Class-XII (2024-25)
Subject- Informatics Practices (065)
APRIL

1. Create a panda's series from a dictionary of values where dictionary having month name as key and monthly income as value.

Solution-

"Create a panda's series from a dictionary of values where dictionary having month name as key and monthly income as value."

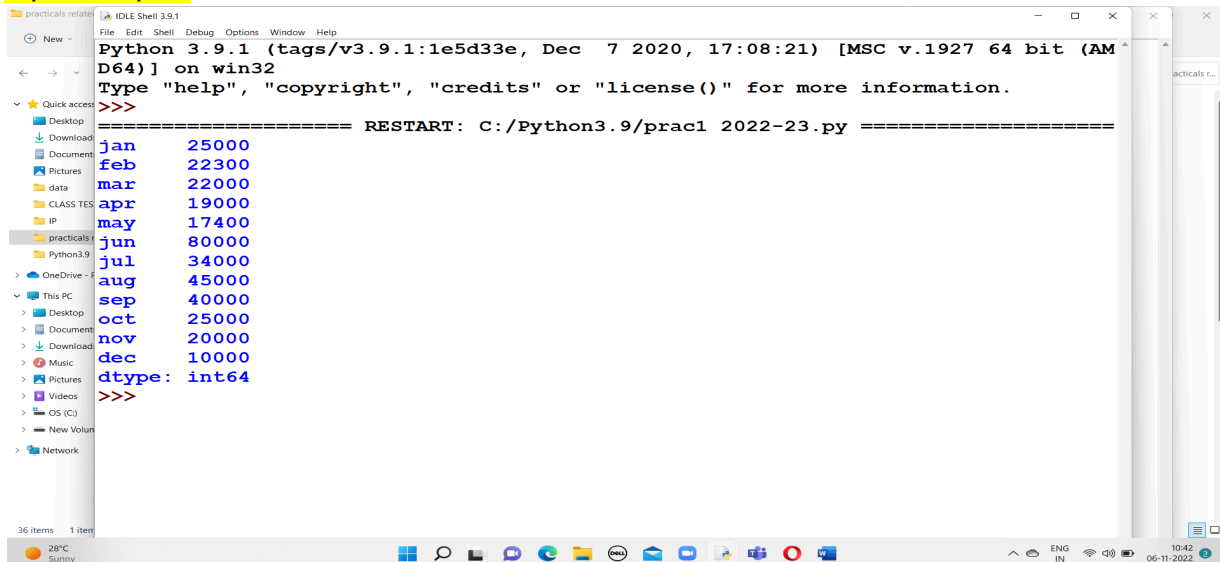
```
import pandas as pd
```

```
D={'jan':25000,'feb':22300,'mar':22000,'apr':19000,'may':17400,'jun':80000,  
'jul':34000,'aug':45000,'sep':40000,'oct':25000,'nov':20000,'dec':10000}
```

```
S=pd.Series(D)
```

```
print(S)
```

Input/Output-



```
Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:/Python3.9/prac1 2022-23.py =====
jan      25000
feb      22300
mar      22000
apr      19000
may      17400
jun      80000
jul      34000
aug      45000
sep      40000
oct      25000
nov      20000
dec      10000
dtype: int64
>>>
```

2. Create a panda's series from a ndarray where ndarray having marks of 5 subjects.

SOLUTION:

```
import pandas as pd
```

```
import numpy as np
```

```
L=[]
```

```
print("Enter Marks of 5 subject : ")
```

```
for i in range(1,6):
```

```
    mark=int(input("Subject"+str(i)+" : "))
```

```
L.append(mark)
```

```
Marks=np.array(L)
```

```
S=pd.Series(Marks)
```

```
print(S)
```

INPUT/OUTPUT:

```

Python 3.9.1 (tags/v3.9.1:1e5d33e, Dec 7 2020, 17:08:21) [MSC v.1927 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
L>>>
===== RESTART: C:/Python3.9/prac2 2022-23.py =====
Enter Marks of 5 subject :
Subject1: 45
Subject2: 44
Subject3: 43
Subject4: 40
Subject5: 39
0 45
1 44
2 43
3 40
4 39
dtype: int32
>>>

```

3. Given a Series of marks, print all the elements that are above 75 marks.

SOLUTION:

```

import pandas as pd
D={'12C01':80, '12C02':60, '12C03':88, '12C04':50, '12C05':60, '12C06':90, '12C07':75}
S=pd.Series(D)
print(S[S>75])

```

INPUT/OUTPUT :

```

===== RESTART: C:/Python3.9/prac2 2022-23.py =====
Enter Marks of 5 subject :
Subject1: 80
Subject2: 56
Subject3: 89
Subject4: 56
Subject5: 75
0 80
2 89
dtype: int32
>>>
===== RESTART: C:/Python3.9/prac3 2022-23.py =====
12C01 80
12C03 88
12C06 90
dtype: int64
>>>

```

JUNE

4. Number of students in classes 11 and 12 in three streams ('Science', 'Commerce' and 'Humanities') are stored in two Series objects c11 and c12. Calculate and display total number of students in classes 11 and 12, stream wise.
02/06/2024

SOLUTION:

```

import pandas as pd
C11=pd.Series(data=[30,40,50],index=['Science','Commerce', 'Humanities'])

```

```
C12=pd.Series(data=[37,44,45],index=['Science','Commerce',
'Humanities'])
print("Total no. of students : ")
print(C11+C12)
```

INPUT/OUTPUT:

```
===== RESTART: C:/Python3.9/prac2 2022-23.py =====
Enter Marks of 5 subject :
Subject1: 80
Subject2: 56
Subject3: 89
Subject4: 56
Subject5: 75
0      80
2      89
dtype: int32
===== RESTART: C:/Python3.9/prac3 2022-23.py =====
12C01      80
12C03      88
12C06      90
dtype: int64
===== RESTART: C:/Python3.9/prac4 2022-23.py =====
Total no. of students :
Science      67
Commerce     84
Humanities   95
dtype: int64
```

5. Create a Series Population and AvgIncome where Population stores the details of population in four metro cities of India and AvgIncome stores the total average income reported in previous year in each of these metros. Calculate income per capita for each of these metro cities.

SOLUTION:

```
import pandas as pd
Population=pd.Series(data=[13059630,14507849,54896570,39058762],\
                    index=['Delhi','Mumbai','Kolkata','Chennai'])
AvgIncome=pd.Series(data=[729865784756,8596756234544,\ 459857586793,\
895859757681], \
                    index=['Delhi','Mumbai','Kolkata','Chennai'])
perCapita=AvgIncome/Population
print('Population in four metro cities')
print(Population)
print('Avg. Income in four metro cities')
print(AvgIncome)
print('Per Capita Income in four metro cities')
print(perCapita)
```

INPUT/OUTPUT:

```
IDLE Shell 3.9.1
File Edit Shell Debug Options Window Help
Science 67
Commerce 84
Humanities 95
dtype: int64
>>>
===== RESTART: C:/Python3.9/prac5 2022-23.py =====
Population in four metro cities
Delhi 13059630
Mumbai 14507849
Kolkata 54896570
Chennai 39058762
dtype: int64
Avg. Income in four metro cities
Delhi 729865784756
Mumbai 8596756234544
Kolkata 459857586793
Chennai 895859757681
dtype: int64
Per Capita Income in four metro cities
Delhi 55887.171747
Mumbai 592558.982006
Kolkata 8376.799986
Chennai 22936.204626
dtype: float64
>>>
```

6. Create a Series that stores the area of some states (atleast 10) in km². Write code to find out the biggest and smallest three areas from the Series. Also print series in ascending order of its value(area).

SOLUTION:

```
import pandas as pd
STAR={'AP':162975,'MH':307713,'CH':135191, 'RJ':342239,'TN':130058,\
      'MP':308245, 'UP':240928,'GJ':196024,'BR':94163, 'KA':191791,\
      'AS':78438,'WB':88752}
Ser1=pd.Series(STAR)
print('States with their area are as follows : \n',Ser1)
print("Top 3 biggest areas are : ")
print(Ser1.sort_values(ascending=False).head(3))
print("Three smallest areas are : ")
print(Ser1.sort_values().head(3))
print('States with Areas (Ascending order of Areas) :')
print(Ser1.sort_values())
```

INPUT/OUTPUT:

States with their area are as follows :

```
AP 162975
MH 307713
CH 135191
RJ 342239
TN 130058
MP 308245
UP 240928
GJ 196024
BR 94163
KA 191791
AS 78438
WB 88752
```

dtype: int64

Top 3 biggest areas are :

```
RJ 342239
MP 308245
MH 307713
```

dtype: int64

Three smallest areas are :

```
AS 78438
```

```
WB 88752
BR 94163
dtype: int64
States with Areas (Ascending order of Areas) :
AS 78438
WB 88752
BR 94163
TN 130058
CH 135191
AP 162975
KA 191791
GJ 196024
UP 240928
MH 307713
MP 308245
RJ 342239
dtype: int64
```

JULY

7. Create a Data Frame quarterly sales where each row contains the item category, item name, and expenditure. Group the rows by the category and print the total expenditure per category. 04-07-2024

SOLUTION:

```
import pandas as pd

Sales_data={"Item_category":["Food','Drink','Food','Sweet','Food','Sweet','Drink'],
'Item_name':['Chole','Pepsi','Paneer','Ladoo','Rajma','Kaju-Katli','Cold Coffee'],
"Expenditure":[350,60,250,200,450,300,150]}
Df_sales=pd.DataFrame(Sales_data,
index=['Item1','Item2','Item3','Item4','Item5','Item6','Item7'])
print(Df_sales)
NewDf=Df_sales.groupby('Item_category')
ExpDf=NewDf['Expenditure'].sum()
print(ExpDf)
```

INPUT/OUTPUT

	Item_category	Item_name	Expenditure
Item1	Food	Chole	350
Item2	Drink	Pepsi	60
Item3	Food	Paneer	250
Item4	Sweet	Ladoo	200
Item5	Food	Rajma	450
Item6	Sweet	Kaju-Katli	300
Item7	Drink	Cold Coffee	150

```
Item_category
Drink    210
Food    1050
Sweet    500
Name: Expenditure, dtype: int64
```

8. Create a data frame for examination result and display row labels, column labels data types of each column and the dimensions.

SOLUTION:

```
import pandas as pd
marks={'Eng':[35,38,40,42,33], 'Acc':[48,43,38,34,30], 'Bst':[37,32,30,37,41],\
      'Eco':[48,43,38,34,30], 'IP':[35,32,31,30,28], 'Total':[188,193,177,193,156],\
      'Percentage':[75.2,77.2,70.8,77.2,62.4]}
result=pd.DataFrame(marks,index=['Rakesh','Sanjay','Neelima','Santosh','Chandni'])
print(result)
print("*****Row Label*****")
print(result.index)
print("*****Column Label*****")
print(result.columns)
print("*****Dimension*****")
print(result.ndim)
print("*****Datatypes of each columns*****")
print(result.dtypes)
print("*****Shape*****")
print(result.shape)
```

INPUT/OUTPUT

```
Eng Acc Bst Eco IP Total Percentage
Rakesh 35 48 37 48 35 188 75.2
Sanjay 38 43 32 43 32 193 77.2
Neelima 40 38 30 38 31 177 70.8
Santosh 42 34 37 34 30 193 77.2
Chandni 33 30 41 30 28 156 62.4
*****Row Label*****
Index(['Rakesh', 'Sanjay', 'Neelima', 'Santosh', 'Chandni'], dtype='object')
*****Column Label*****
Index(['Eng', 'Acc', 'Bst', 'Eco', 'IP', 'Total', 'Percentage'], dtype='object')
*****Dimension*****
2
*****Datatypes of each columns*****
Eng      int64
Acc      int64
Bst      int64
Eco      int64
IP       int64
Total    int64
Percentage float64
dtype: object
*****Shape*****
(5, 7)
```

9. Write a program to create a Data Frame Quarterly Sales where each row contains the Quarterly Sales of TV, Freeze and AC. Show the DataFrame after deleting details of Freeze.

05-07-2024

SOLUTION:

```
import pandas as pd
import numpy as np
s1=pd.Series([280,400,976], index=['TV','Freeze','AC'])
s2=pd.Series([300,233,198],index=['TV','Freeze','AC'])
s3=pd.Series([323,315,206],index=['TV','Freeze','AC'])
s4=pd.Series([279,722,298],index=['TV','Freeze','AC'])
df=pd.DataFrame([s1,s2,s3,s4],index=['QTR1','QTR2','QTR3','QTR4'])
print("Quarterly Sales : ")
```

```
print(df)
df=df.drop(['Freeze'],axis=1)
print("Quarterly Sales without detail of Freeze : ")
print(df)
```

INPUT/OUTPUT:

```
===== RESTART: C:/Python3.9/prac7 2022-23.py :
Quarterly Sales :
      TV  Freeze   AC
QTR1  280     400  976
QTR2  300     233  198
QTR3  323     315  206
QTR4  279     722  298
Quarterly Sales without detail of Freeze :
      TV   AC
QTR1  280  976
QTR2  300  198
QTR3  323  206
QTR4  279  298
```

10. Create a data frame of students' marks with name of student as label of index and subject as column index. Display subject wise sum of marks.

SOLUTION:

```
import pandas as pd
import numpy as np
s1=pd.Series([28,40,36, 44, 39], index=['ENG','ACC','BST', 'ECO', 'IP'])
s2=pd.Series([30,33,19,23,38],index=['ENG','ACC','BST', 'ECO', 'IP'])
s3=pd.Series([32,31,20, 34,45],index=['ENG','ACC','BST', 'ECO', 'IP'])
s4=pd.Series([27,22,29,44,38],index=['ENG','ACC','BST', 'ECO', 'IP'])
s5=pd.Series([20,43,36, 44, 39], index=['ENG','ACC','BST', 'ECO', 'IP'])
s6=pd.Series([30,33,29,23,38],index=['ENG','ACC','BST', 'ECO', 'IP'])
s7=pd.Series([32,31,26, 34,45],index=['ENG','ACC','BST', 'ECO', 'IP'])
s8=pd.Series([27,42,29,44,28],index=['ENG','ACC','BST', 'ECO', 'IP'])
df=pd.DataFrame([s1,s2,s3,s4,s5,s6,s7,s8],index=['ABHI','BIMLESH','CHETAN',\
        'DINESH','CHETNA','VISHAL','RAKESH','MUKESH'])

print(df)
print("Subject wise sum of marks are : ")
print(df.sum())
```

INPUT/OUTPUT:

```
      ENG  ACC  BST  ECO  IP
ABHI    28   40   36   44   39
BIMLESH  30   33   19   23   38
CHETAN   32   31   20   34   45
DINESH   27   22   29   44   38
CHETNA   20   43   36   44   39
VISHAL   30   33   29   23   38
RAKESH   32   31   26   34   45
MUKESH   27   42   29   44   28
Subject wise sum of marks are :
ENG      226
ACC      275
BST      224
ECO      290
IP       310
dtype: int64
```

11. Create a dataframe of students' marks with name of student as label of index and subject as column index. Calculate and display student wise marks and percentage scored.
14-07-2024

SOLUTION:

```
import pandas as pd
#Creating Data Frame
s1=pd.Series([28,40,36, 44, 39], index=['ENG','ACC','BST', 'ECO', 'IP'])
s2=pd.Series([30,33,19,23,38],index=['ENG','ACC','BST', 'ECO', 'IP'])
s3=pd.Series([32,31,20, 34,45],index=['ENG','ACC','BST', 'ECO', 'IP'])
s4=pd.Series([27,22,29,44,38],index=['ENG','ACC','BST', 'ECO', 'IP'])
s5=pd.Series([20,43,36, 44, 39], index=['ENG','ACC','BST', 'ECO', 'IP'])
df=pd.DataFrame([s1,s2,s3,s4,s5],index=['ABHI','BIMLESH','CHETAN','DINESH','CHETNA'])
print(df)
#Displaying Student wise marks and percentage scored
for (name,marks) in df.iterrows():
    TotalM=sum(marks)
per=TotalM/2.5
print("Marks scored by",name,"are :")
print("Eng:",marks[0],"Acc.:",marks[1],"BST :",marks[2],"Eco :",marks[3],"IP :",marks[4])
print('Total Marks : ',TotalM,"Percentage Scored : ",per)
```

INPUT/OUTPUT:

```
      ENG  ACC  BST  ECO  IP
ABHI    28   40   36   44   39
BIMLESH  30   33   19   23   38
CHETAN   32   31   20   34   45
DINESH   27   22   29   44   38
CHETNA   20   43   36   44   39
Marks scored by ABHI are :
Eng: 28 Acc.: 40 BST : 36 Eco : 44 IP : 39
Total Marks : 187 Percentage Scored : 74.8
Marks scored by BIMLESH are :
Eng: 30 Acc.: 33 BST : 19 Eco : 23 IP : 38
Total Marks : 143 Percentage Scored : 57.2
Marks scored by CHETAN are :
Eng: 32 Acc.: 31 BST : 20 Eco : 34 IP : 45
Total Marks : 162 Percentage Scored : 64.8
Marks scored by DINESH are :
Eng: 27 Acc.: 22 BST : 29 Eco : 44 IP : 38
Total Marks : 160 Percentage Scored : 64.0
Marks scored by CHETNA are :
Eng: 20 Acc.: 43 BST : 36 Eco : 44 IP : 39
Total Marks : 182 Percentage Scored : 72.8
```

12. Create the following DataFrame Sales containing year wise sales figures for five salespersons in INR. Use the years as column labels, and salesperson names as row labels.

	2018	2019	2020	2021
Kapil	110	205	177	189
Kamini	130	165	175	190
Shikhar	115	206	157	179

Mohini	118	198	183	169
--------	-----	-----	-----	-----

- Create the DataFrame.
- Display the row labels of Sales.
- Display the column labels of Sales.
- Display the data types of each column of Sales.
- Display the dimensions, shape, size and values of Sales.

SOLUTION:

```
import pandas as pd
#a)Creating Data Frame
s1=pd.Series([110,205,177,189],index=['2018','2019','2020','2021'])
s2=pd.Series([130,165,175,190],index=['2018','2019','2020','2021'])
s3=pd.Series([115,206,157,179],index=['2018','2019','2020','2021'])
s4=pd.Series([118,198,183,169],index=['2018','2019','2020','2021'])
df=pd.DataFrame([s1,s2,s3,s4],index=['Kapil','Kamini','Shikhar','Mohini'])
print(df)
#b)Display the row labels of Sales
print('Row Labels are : ', df.index.values)
#c)Display the column labels of Sales.
print('Column Labels are : ', df.columns.values)
#d)Display the data types of each column of Sales.
print('Datatype of each column are:\n', df.dtypes)
#e)Display the dimensions, shape, size and values of Sales.
print('No. of Dimensions:',df.ndim,'\nShape:',df.shape,'\nSize:',df.size,'\nValues:',df.values)
```

INPUT/OUTPUT:

```
===== RESTART: C:/Python3.9/prac10 2022-23.py
      2018   2019   2020   2021
Kapil    110    205    177    189
Kamini    130    165    175    190
Shikhar   115    206    157    179
Mohini    118    198    183    169
Row Labels are :   ['Kapil' 'Kamini' 'Shikhar' 'Mohini']
Column Labels are :   ['2018' '2019' '2020' '2021']
Datatype of each column are:
2018      int64
2019      int64
2020      int64
2021      int64
dtype: object
No. of Dimensions: 2
Shape: (4, 4)
Size: 16
Values: [[110 205 177 189]
 [130 165 175 190]
 [115 206 157 179]
 [118 198 183 169]]
```

AUGUST

- Consider above dataframe and write code to do the following:
 - Display the last two rows of Sales.
 - Display the first two columns of Sales.

SOLUTION:

```

import pandas as pd
#Creating Data Frame
s1=pd.Series([110,205,177,189],index=['2018','2019','2020','2021'])
s2=pd.Series([130,165,175,190],index=['2018','2019','2020','2021'])
s3=pd.Series([115,206,157,179],index=['2018','2019','2020','2021'])
s4=pd.Series([118,198,183,169],index=['2018','2019','2020','2021'])
df=pd.DataFrame([s1,s2,s3,s4],index=['Kapil','Kamini','Shikhar','Mohini'])
print("Sales Detail\n",df)
#a)Display the last two rows of Sales.
print("Last two rows of Sales detail as follows : ")
print(df.tail(2))
#b)Display the first two columns of Sales.
print("First two columns of Sales detail as follows : ")
print(df.loc[:, '2018':'2019'])

```

INPUT/OUTPUT:

```

===== RESTART: C:/Python3.9/prac11 2022-23.py
Sales Detail
      2018  2019  2020  2021
Kapil   110   205   177   189
Kamini  130   165   175   190
Shikhar 115   206   157   179
Mohini  118   198   183   169
Last two rows of Sales detail as follows :
      2018  2019  2020  2021
Shikhar 115   206   157   179
Mohini  118   198   183   169
First two columns of Sales detail as follows :
      2018  2019
Kapil   110   205
Kamini  130   165
Shikhar 115   206
Mohini  118   198

```

14. Use above dataframe and do the following:

- Change the DataFrame Sales such that it becomes its transpose.
- Display the sales made by all sales persons in the year 2018.
- Display the sales made by Kapil and Mohini in the year 2019 and 2020.
- Add data to Sales for salesman Nirali where the sales made are [221, 178, 165, 177, 210] in the years [2018, 2019, 2020, 2021] respectively
- Delete the data for the year 2018 from the DataFrame Sales.
- Delete the data for sales man Shikhar from the DataFrame Sales.
- Change the name of the salesperson Kamini to Rani and Kapil to Anil.
- Update the sale made by Mohini in 118 to 150 in 2018.

SOLUTION:

```

import pandas as pd
#Creating Data Frame
s1=pd.Series([110,205,177,189],index=['2018','2019','2020','2021'])
s2=pd.Series([130,165,175,190],index=['2018','2019','2020','2021'])
s3=pd.Series([115,206,157,179],index=['2018','2019','2020','2021'])
s4=pd.Series([118,198,183,169],index=['2018','2019','2020','2021'])
df=pd.DataFrame([s1,s2,s3,s4],index=['Kapil','Kamini','Shikhar','Mohini'])
print("Sales Detail\n",df)

```

```

#a)Change the DataFrame Sales such that it becomes its transpose.
print("Transpose of Sales : ")
print(df.T)
#b)Display the sales made by all sales persons in the year 2018.
print("Sales made by salesperson in year 2018 : ")
print(df['2018'])
#c)Display the sales made by Kapil and Mohini in the year 2019 and 2020.
print("Sales made by Kapil and Mohini in 2019 and 2020 : ")
df1=df[['2019','2020']]
print(df1.loc['Kapil':'Kapil',:])
print(df1.loc['Mohini':'Mohini',:])
#d)Add data to Sales for salesman Nirali where the sales made are
#[221, 178, 165, 177] in the years [2018, 2019, 2020, 2021] respectively
print("Sales detail after adding one more row of Salesperson Nirali :")
df.loc['Nirali',:]=[221,178,165,177]
print(df)
#e)Update the sale made by Mohini in 118 to 150 in 2018.
df.loc['Mohini','2018']=150
print("Sales detail after modifying data of Mohini for year 2018:")
print(df)
#f)Delete the data for the year 2018 from the DataFrame Sales.
df=df.drop(['2018'],axis=1)
print("Sales detail after deleting data for year 2018:")
print(df)
#g)Delete the data for salesman Shikhar from the DataFrame Sales.
df=df.drop(['Shikhar'])
print("Sales detail after deleting data of Shikhar:")
print(df)
#h)Change the name of the salesperson Kamini to Rani and Kapil to Anil.
df.rename({'Kamini':'Rani','Kapil':'Anil'},inplace=True)
print("Sales detail with modified salesperson name :")
print(df)

```

INPUT/OUTPUT:

```
===== RESTART: C:\Python3.9\prac12 2022-23.py
```

```
Sales Detail
```

	2018	2019	2020	2021
Kapil	110	205	177	189
Kamini	130	165	175	190
Shikhar	115	206	157	179
Mohini	118	198	183	169

```
Transpose of Sales :
```

	Kapil	Kamini	Shikhar	Mohini
2018	110	130	115	118
2019	205	165	206	198
2020	177	175	157	183
2021	189	190	179	169

```
Sales made by salesperson in year 2018 :
```

Kapil	110
Kamini	130
Shikhar	115
Mohini	118

```
Name: 2018, dtype: int64
```

```
Sales made by Kapil and Mohini in 2019 and 2020 :
```

	2019	2020
Kapil	205	177
	2019	2020
Mohini	198	183

```
Sales detail after adding one more row of Salesperson Nirali :
```

	2018	2019	2020	2021
Kapil	110.0	205.0	177.0	189.0
Kamini	130.0	165.0	175.0	190.0
Shikhar	115.0	206.0	157.0	179.0
Mohini	118.0	198.0	183.0	169.0
Nirali	221.0	178.0	165.0	177.0

```
Sales detail after modifying data of Mohini for year 2018:
```

	2018	2019	2020	2021
Kapil	110.0	205.0	177.0	189.0
Kamini	130.0	165.0	175.0	190.0
Shikhar	115.0	206.0	157.0	179.0
Mohini	150.0	198.0	183.0	169.0
Nirali	221.0	178.0	165.0	177.0

```
Sales detail after deleting data for year 2018:
```

	2019	2020	2021
Kapil	205.0	177.0	189.0
Kamini	165.0	175.0	190.0
Shikhar	206.0	157.0	179.0
Mohini	198.0	183.0	169.0
Nirali	178.0	165.0	177.0

```
Sales detail after deleting data of Shikhar:
```

	2019	2020	2021
Kapil	205.0	177.0	189.0
Kamini	165.0	175.0	190.0
Mohini	198.0	183.0	169.0
Nirali	178.0	165.0	177.0

```
Sales detail with modified salesperson name :
```

	2019	2020	2021
Anil	205.0	177.0	189.0
Rani	165.0	175.0	190.0
Mohini	198.0	183.0	169.0
Nirali	178.0	165.0	177.0

15. Write a program to input and create a DataFrame to store Roll Number, Name and percentage of five students and write above data frame in a csv file "marks.csv".

```
import pandas as pd
```

$$L = []$$

```
for i in range(1,6):
```

```
rn=int(input('Enter Roll no. of student '+str(i)+' : '))
```

```
nm=input('Enter Name of student '+str(i)+' :')
```

```
mks=int(input('Enter Marks of student '+str(i)+' : '))
```

$$L1=[rn,nm,mks]$$

```
L.append(L1)
```

```
df=pd.DataFrame(L,columns=['Rollno','Name','Marks'])
```

```
print('Data Frame for the student detail as follows:')

```

```
print(df)
```

```
df.to_csv('C:\\Python3.9\\marks.csv')
```

```
print('marks.csv file created successfully!')
```

INPUT/OUTPUT:

```
===== RESTART: C:\Python3.9\prac14 2022-23.py
```

Enter Roll no. of student 1 : 101

Enter Name of student 1 : abc

Enter Marks of student 1 : 23

Enter Roll no. of student 2 : 102

Enter Name of student 2 : bbb

Enter Marks of student 2 : 33

Enter Roll no. of student 3 : 103

Enter Name of student 3 : ccc

Enter Marks of student 3 : 34

Enter Roll no. of student 4 : 104

Enter Name of student 4 : ddd

Enter Marks of student 4 : 39

Enter Roll no. of student 5 : 105

Enter Name of student 5 : eee

Enter Marks of student 5 : 41

Data Frame for the student detail as follows:

Rollno	Name	Marks
--------	------	-------

0	101	abc	23
---	-----	-----	----

```
1      102  bbb      33
```

2 103 ccc 34

3 104 ddd 39

4 105 eee 41

```
marks.csv file created successfully!
```

16. Write a program to read “marks.csv” file and display student details in the following format-

Roll no: XXXXX

Name: abcd

Percentage Score: XXX.XX

1

Total Student:XX

Average Percentage:XXX.XX

[illegible]

SOLUTION:

```
import pandas as pd
df=pd.read_csv('C:\\Python3.9\\marks.csv',header=None,skiprows=1,names=['Roll','Name','Marks'])
ts=0
print(df)
for (row,data) in df.iterrows():
    print('Roll no:',data[0])
    print('Name:',data[1])
    print('Percentage Scored:',data[2])
    ts=ts+1
print('Total Student:',ts)
print('Average Percentage:',df['Marks'].sum()/ts)
```

INPUT/OUTPUT:

```
===== RESTART: C:/Python3.9/prac15 2022-23.py
Roll Name Marks
0 101 aaa 23
1 102 bbb 24
2 103 ccc 25
3 104 ddd 26
4 105 eee 21
Roll no: 101
Name: aaa
Percentage Scored: 23
Roll no: 102
Name: bbb
Percentage Scored: 24
Roll no: 103
Name: ccc
Percentage Scored: 25
Roll no: 104
Name: ddd
Percentage Scored: 26
Roll no: 105
Name: eee
Percentage Scored: 21
Total Student: 5
Average Percentage: 23.8
```

17. Plot the following data on a line chart and customize the chart according to the below-given instructions:

Sales Report

Month	January	February	March	April	May
Sales	510	350	475	580	600

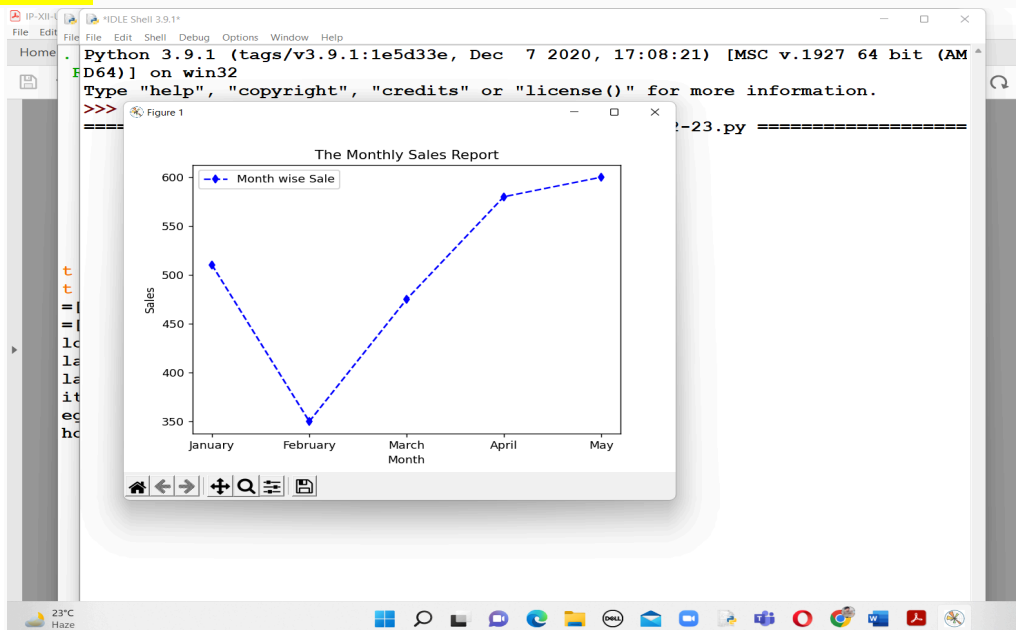
1. Write a title for the chart "The Monthly Sales Report"
2. Write the appropriate titles of both the axes
3. Write code to Display legends
4. Display blue color for the line
5. Use the line style – dashed

6. Display diamond style markers on data points.

SOLUTION:

```
import pandas as pd
import matplotlib.pyplot as plt
Month=['January','February','March','April','May']
Sales=[510,350,475,580,600]
plt.plot(Month,Sales,linestyle='dashed',marker='d',color='b',label='Month wise Sale')
plt.xlabel('Month')
plt.ylabel('Sales')
plt.title('The Monthly Sales Report')
plt.legend()
plt.show()
```

OUTPUT:



18. Pratyush Garments has recorded the following data into their register for their income from cotton clothes and jeans. Plot them on the line chart.

Day	Monday	Tuesday	Wednesday	Thursday	Friday
Cotton	450	560	400	605	580
Jeans	490	600	425	610	625

Apply following customization to the line chart.

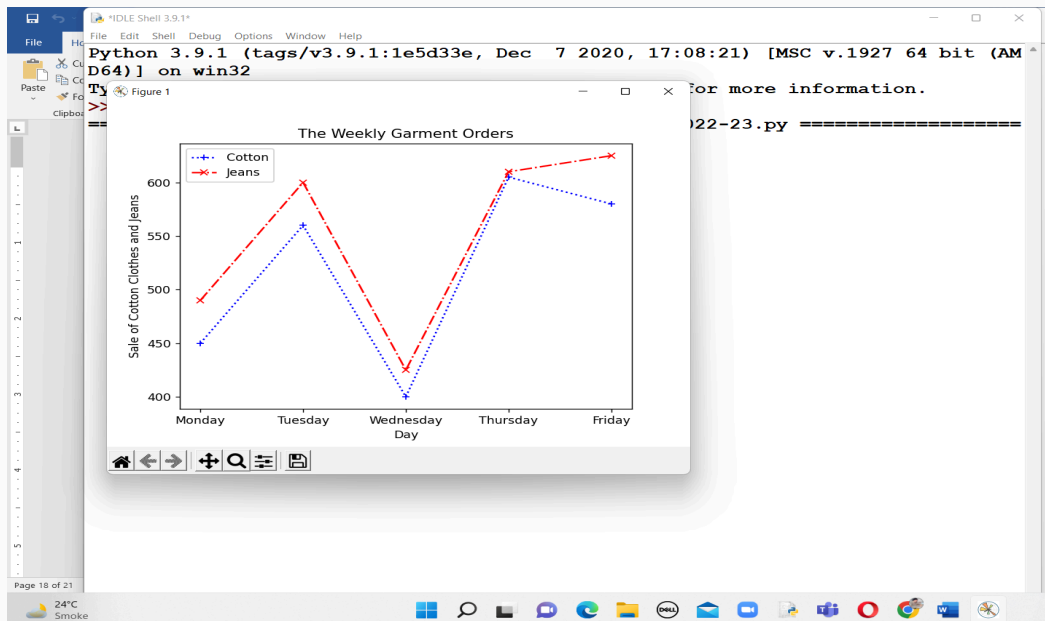
1. Write a title for the chart "The Weekly Garment Orders".
2. Write the appropriate titles of both the axes.
3. Write code to Display legends.
4. Display your choice of colors for both the lines cotton and jeans.
5. Use the line style – dotted for cotton and dashdot for jeans.
6. Display plus markers on cotton and x markers of jeans.

SOLUTION:

```
import pandas as pd
import matplotlib.pyplot as plt
weekday=['Monday','Tuesday','Wednesday','Thursday','Friday']
Cotton=[450,560,400,605,580]
Jeans=[490,600,425,610,625]
```

```
plt.plot(weekday,Cotton,linestyle='dotted',marker='+',color='b',label='Cotton')
plt.plot(weekday,Jeans,linestyle='dashdot',marker='x',color='r',label='Jeans')
plt.xlabel('Day')
plt.ylabel('Sale of Cotton Clothes and Jeans')
plt.title('The Weekly Garment Orders')
plt.legend()
plt.show()
```

INPUT/OUTPUT:



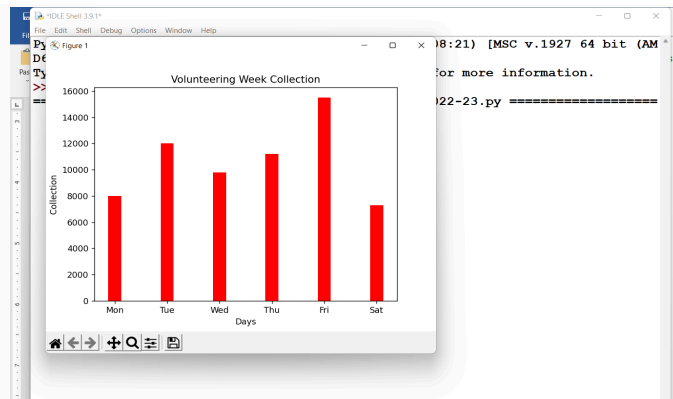
SEPTEMBER

19. TSS school celebrated volunteering week where each section of class XI dedicated a day for collecting amount for charity being supported by the school. Section A volunteered on Monday, B on Tuesday, C on Wednesday and so on. There are six sections in class XI. Amounts collected by sections A to F are 8000, 12000, 9800, 11200, 15500, 7300.
- Write a program to create a bar chart showing collection amount.
 - The graph should have proper title and axes titles.
 - The ticks on X-axis should have Day Names.

SOLUTION:

```
import pandas as pd
import matplotlib.pyplot as plt
weekday=['Mon','Tue','Wed','Thu','Fri','Sat']
Coll=[8000, 12000, 9800, 11200, 15500, 7300]
Jeans=[490,600,425,610,625]
plt.bar(weekday,Coll,color='r', width=0.25)
plt.xlabel('Days')
plt.ylabel('Collection')
plt.title('Volunteering Week Collection')
plt.xticks(weekday)
plt.show()
```

INPUT/OUTPUT:



20. Display comparison of Run Scored by two teams in a cricket match using bar charts on the same plane with following features:

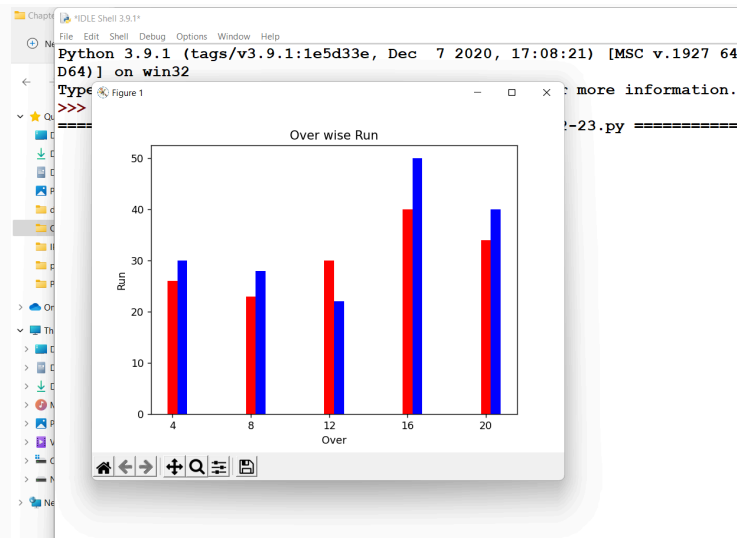
Over	RunTeamA	RunTeamB
5	26	30
10	23	28
15	30	22
20	50	40

- Write a program to create a bar chart showing bars for teamA and teamB side by side.
- Use red color for bars of TeamA and blue for bars of TeamB
- The graph should have proper title and axes titles.
- The ticks on X-axis should have Overs.

SOLUTION:

```
import numpy as np
import matplotlib.pyplot as plt
Over=np.array([4,8,12,16,20])
RunTeamA=[26,23,30,40,34]
RunTeamB=[30,28,22,50,40]
plt.bar(Over,RunTeamA,color='r', width=0.5)
plt.bar(Over+0.5,RunTeamB,color='b', width=0.5)
plt.xlabel('Over')
plt.ylabel('Run')
plt.title('Over wise Run')
plt.xticks(Over)
plt.show()
```

INPUT/OUTPUT:



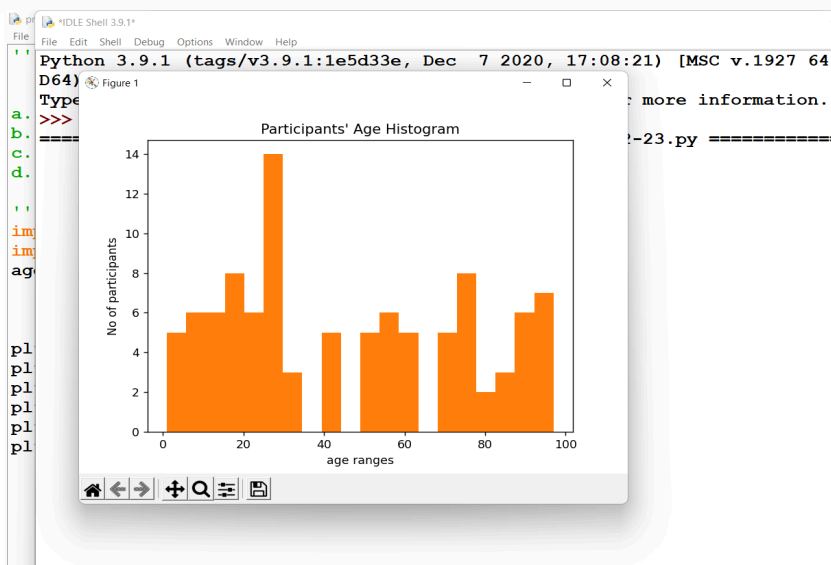
21. A survey gathers height and weight of 100 participants and recorded the participants' ages (assume any 100 values, each value must be between 1 to 100, you can repeat values as many times you wish).

Write a program to plot a histogram from above data with 20 bins.

SOLUTION:

```
import numpy as np
import matplotlib.pyplot as plt
ages=[1,1,2,3,4,7,8,9,10,10,10,11,13,13,13,13,15,16,17,18,19,19,20,20,20,\
      23,23,24,24,24,24,25,25,25,25,25,25,25,27,27,27,28,28,29,30,30,30,40,\
      40,42,42,43,50,50,50,51,52,55,57,57,58,58,58,60,60,60,63,63,70,70,70,70,\
      70,73,73,73,73,74,76,76,77,79,80,83,85,85,88,89,89,89,90,91,94,94,94,94,96,97,97]
plt.hist(ages, bins=20)
plt.hist(ages, bins=20)
plt.xlabel('age ranges')
plt.ylabel('No of participants')
plt.title('Participants\' Age Histogram')
plt.show()
```

INPUT/OUTPUT:



OCTOBER

Q.22.Consider the following table PRODUCT-

PRODUCT

PCODE WARRANTY	PNAME	COMPANY	PRICE	STOCK	MANUFACTURE
P001	TV 3	BPL	10000	200	12-JAN-2008
P002	TV 4	SONY	12000	150	23-MAR-2007
P003	PC 2	LENOVO	39000	100	09-APR-2008
P004	PC 2	COMPAQ	38000	120	20-JUN-2009
P005	HANDYCAM 3	SONY	18000	250	23-MAR-2007

(a) Write MySQL commands for following Statements

- To show details of all PC with stock more than 110.
- To list the company which gives warranty for more than 2 years?
- To find stock value of the BPL Company where stock value is sum of the products of price and stock.
- To show number of products from each company.
- To count the number of PRODUCTS which are manufactured in 2009?
- To show the PRODUCT name which are within warranty as on date?

(b) Give the output of following MySQL statement.

- Select COUNT (distinct company) from PRODUCT;
- Select MAX (price) from PRODUCT where WARRANTY<=3;
- Select AVG(price) from PRODUCT where Company="SONY";
- Select MIN(price) from PRODUCT where stock<200;

ANSWER:

(a)

- SELECT * FROM PRODUCT WHERE PNAME='PC' AND STOCK>110;
- SELECT COMPANY FROM PRODUCT WHERE WARRANTY>2;
- SELECT SUM(STOCK*PRICE) AS "STOCK VALUE" FROM PRODUCT WHERE COMPANY='BPL';
- SELECT COMPANY,COUNT(*) "NO OF PRODUCT" FROM PRODUCT GROUP BY COMPANY;
- SELECT COUNT(*) "NO OF PRODUCT MANUFACTURED IN 2009" FROM PRODUCT WHERE YEAR(MANUFACTURE)=2009;
- SELECT PNAME FROM PRODUCT WHERE TRUNCATE(DATEDIFF(CURDATE(),MANUFACTURE)/ 365,0)>WARRANTY;

(b)

(i)

```
+-----+
| COUNT(distinct company) |
+-----+
|                          4 |
+-----+
```

(ii)

```
+-----+
| MAX(price) |
+-----+
|          39000 |
+-----+
```

(iii)

```

+-----+
| AVG(price) |
+-----+
| 15000.0000 |
+-----+

```

(iv)

```

+-----+
| MIN(price) |
+-----+
| 12000 |
+-----+

```

Q.23. Study the following table STUDENT and write MySQL command for the questions (i) to (vii)

TABLE: STUDENT

SID	Name	Stream	Gender	Marks
101	Siddharat	Science	M	42
104	Raghav	Science	M	26
107	Naman	Commerce	M	40
114	Nupur	Humanities	F	13
109	Janvi	Commerce	F	19
105	Rama	Humanities	M	30
117	James	Science	F	23
111	Binoy	Science	F	12
130	Samuel	Commerce	M	45

- Create a student table with the student id, name, Stream, gender and marks as attributes where the student id is the primary key.
- Insert the details of students as given in the above table.
- Delete the details of students whose name starts with letter 'B'.
- Display the details of the students with marks more than 80.
- Find the min, max, sum, and average of the marks in a student marks table.
- Find the total number of students stream wise.
- Display the Name and Marks of Students in descending order of the marks.

SOLUTION:

- CREATE TABLE STUDENT(SID INT NOT NULL PRIMARY KEY, NAME VARCHAR(20), STREAM VARCHAR(20), GENDER CHAR(1), MARKS INT);
- INSERT INTO STUDENT VALUES(101,'Siddharat','Science','M',42);
INSERT INTO STUDENT VALUES(104,'Raghav','Science','M',26);
INSERT INTO STUDENT VALUES(107,'Naman','Commerce','M',40);
INSERT INTO STUDENT VALUES(114,'Nupur','Humanities','F',13);
INSERT INTO STUDENT VALUES(109,'Janvi','Commerce','F',19);
INSERT INTO STUDENT VALUES(105,'Rama','Humanities','M',30);
INSERT INTO STUDENT VALUES(117,'James','Science','F',23);
INSERT INTO STUDENT VALUES(111,'Binoy','Science','F',12);
INSERT INTO STUDENT VALUES(130,'Samuel','Commerce','M',45);
- Delete FROM STUDENT WHERE NAME LIKE "B%";
- SELECT * FROM STUDENT WHERE MARKS>80;
- SELECT MIN(MARKS), MAX(MARKS), SUM(MARKS), AVG(MARKS) FROM STUDENT;
- SELECT STREAM, COUNT(*) FROM STUDENT GROUP BY STREAM;
- SELECT NAME, MARK FROM STUDENT ORDER BY MARKS DESC;

Q.24. Answer the following questions based on the table CLUB given below:

TABLE :CLUB

Column Name	Data Type	Size	Constraints	Description
Member_No	Number	5	Primary Key	Member number
Member_Name	Varchar2	40	Not Null	Name of the member
Address	Varchar2	30		Address of he member
Age	Number	2	>=18	Age of the member
Type	Varchar2	10		Membershiptype(Temp or Permanent)
Fees	Number	6,2		Membership fees

- Write the SQL command to create the table CLUB including the constraints.
- Insert two tuples.
- Write the SQL command to display the details of all the members whose type is "Permanent" and fees is more than Rs. 5000.
- Write SQL query to add a new column called Phno.
- Change the fees rate by 10% if the membership is of type "Temp"
- Write the SQL command to display all the details of all the members whose age is greater than eighteen.

Write the output of the following SQL queries:

- SELECT ROUND(6.5675, 2);
- SELECT TRUNCATE(5.3456, 1);
- SELECT DAYOFMONTH('2009-08-25');
- SELECT MID('Class 12', 2,3);

ANSWER:

- CREATE TABLE CLUB(Member_No int(5) not null primary key, Member_Name varchar(40) not null, Address varchar(30), Age int(2), Type varchar(10), Fees Numeric(6,2), check (Age>=18));

```
mysql> Insert into CLUB Values(10001, 'Atul', 'Dhoomanganj Prayagraj', 20, 'Temp', 1050.00);
Query OK, 1 row affected (0.00 sec)

mysql> Insert into CLUB Values(10002, 'Zishan', 'Civillines Prayagraj', 22, 'Permanent', 3050.00);
Query OK, 1 row affected (0.03 sec)
```

- SELECT * FROM CLUB WHERE TYPE='PERMANENT' AND FEES>5000;
- ALTER TABLE CLUB ADD PHNO INT;
- UPDATE CLUB SET FEES=FEES*0.1 WHERE MEMBERSHIP='TEMP';
- SELECT * FROM CLUB WHERE AGE>18;

OUTPUTS

```
mysql> SELECT ROUND(6.5675,2);
+-----+
| ROUND(6.5675,2) |
+-----+
|                6.57 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT TRUNCATE(5.3456,1);
+-----+
| TRUNCATE(5.3456,1) |
+-----+
|                5.3 |
+-----+
1 row in set (0.00 sec)
```

```
mysql> SELECT DAYOFMONTH('2009-08-25');
+-----+
| DAYOFMONTH('2009-08-25') |
+-----+
| 25 |
+-----+
1 row in set (0.00 sec)

mysql> SELECT MID('cLASS 12',2,3);
+-----+
| MID('cLASS 12',2,3) |
+-----+
| LAS |
+-----+
1 row in set (0.00 sec)
```

Q.25. Consider the table TEACHER given below. Write commands in SQL for (i) to (iv) and output for (v) to (viii)

TEACHER

ID	Name	Department	Hiredate	Category	Gender	Salary
1	Tanya Nanda	SocialStudies	3/17/1994	TGT	F	25000
2	Saurabh Sharma	Art	2/12/1990	PRT	M	20000
3	Nandita Arora	English	5/16/1980	PGT	F	30000
4	James Jacob	English	10/16/1989	TGT	M	25000
5	Jaspreet Kaur	Hindi	8/1/1990	PRT	F	22000
6	Disha Sehgal	Math	3/17/1980	PRT	F	21000
7	Siddharth Kapoor	Science	9/2/1994	TGT	M	27000
8	Sonali Mukherjee	Math	11/17/1980	TGT	F	24500

- i. Display details of oldest teacher.
- ii. Display total salary of Female employee Category wise.
- iii. To list names, departments and date of hiring of all the teachers in ascending order of date of joining
- iv. To count the number of teachers department wise.
- v. SELECT MAX(Hiredate) FROM Teacher;
- vi. SELECT DISTINCT(category) FROM teacher;
- vii. SELECT COUNT(*) FROM TEACHER WHERE Category = "PGT"
- viii. SELECT AVG(Salary) FROM TEACHER group by Gender;

SOLUTIONS:

```
mysql> SELECT * FROM TEACHER WHERE HIREDATE=(SELECT MIN(HIREDATE) FROM TEACHER);
+-----+-----+-----+-----+-----+-----+-----+
| ID | NAME | DEPARTMENT | HIREDATE | CATEGORY | GENDER | SALARY |
+-----+-----+-----+-----+-----+-----+-----+
| 6 | DISHA SEHGAL | MATH | 1980-03-17 | PRT | F | 21000.00 |
+-----+-----+-----+-----+-----+-----+-----+
```

```
mysql> SELECT GENDER, SUM(SALARY) FROM TEACHER GROUP BY GENDER HAVING GENDER='F';
```

GENDER	SUM(SALARY)
F	122500.00

```
1 row in set (0.00 sec)
```

```
mysql> SELECT NAME, DEPARTMENT, HIREDATE FROM TEACHER ORDER BY HIREDATE;
```

NAME	DEPARTMENT	HIREDATE
DISHA SEHGAL	MATH	1980-03-17
NANDITA ARORA	ENGLISH	1980-05-16
SONALI MUKHERJEE	MATH	1980-11-17
JAMES JACOB	ENGLISH	1989-10-16
SAURABH SHARMA	ART	1990-02-12
JASPREET KAUR	HINDI	1990-08-01
TANYA NANDA	SOCIALSTUDIES	1994-03-17
SIDDHARTH KAPOOR	SCIENCE	1994-09-02

```
8 rows in set (0.00 sec)
```

```
mysql> SELECT DEPARTMENT, COUNT(DEPARTMENT) FROM TEACHER GROUP BY DEPARTMENT;
```

DEPARTMENT	COUNT(DEPARTMENT)
ART	1
ENGLISH	2
HINDI	1
MATH	2
SCIENCE	1
SOCIALSTUDIES	1

```
mysql> SELECT MAX(HIREDATE) FROM TEACHER;
```

MAX(HIREDATE)
1994-09-02

```
1 row in set (0.00 sec)
```

```
mysql> SELECT DISTINCT(CATEGORY) FROM TEACHER;
```

CATEGORY
TGT
PRT
PGT

```
3 rows in set (0.00 sec)
```

```
mysql> SELECT COUNT(*) FROM TEACHER WHERE CATEGORY='PGT';
```

COUNT(*)
1

```
1 row in set (0.00 sec)
```

```
mysql> SELECT AVG(SALARY) FROM TEACHER GROUP BY GENDER;
```

AVG(SALARY)
24500.000000
24000.000000