Name:Ayush Bhalerao

Std: XII Div: Science CBSE

Board Seat No._____

Subject: Informatics Practices

Practical File

School Roll No.:05

Year: 2025-26

City Pride School

Pradhikaran, Nigdi-411044

INDEX
Python Programs

INDEX
SQL Queries

PYTHON PROGRAMS

1) Create panda series from a dictionary of values and ndarray.

CODE:

```
import pandas as pd
import numpy as np
d1={'Ram':67,'lakshman':45,'Sita':78,'Krishna':88,'Radha':81,'Rukmani':65
    ,'Ravan':53,'Hanuman':64,'Shivan':99,'Parvathi':95}
s1=pd.Series(d1)
print(s1)
a=np.array([10,20,30,40,50])
print('\n')
s2=pd.Series(a)
print(s2)
```

OUTPUT:

```
Ram          67
lakshman     45
Sita         78
Krishna      88
Radha        81
Rukmani      65
Ravan        53
Hanuman      64
Shivan       99
Parvathi     95
dtype: int64


0    10
1    20
2    30
3    40
4    50
dtype: int64
```

2) In a given series, print all the elements that are above 75th percentile.

CODE:

```
import pandas as pd
d1={'Ram':67,'lakshman':45,'Sita':78,'Krishna':88,'Radha':81,'Rukmani':65
    ,'Ravan':53,'Hanuman':64,'Shivan':99,'Parvathi':95}
s1=pd.Series(d1)
print(s1[s1>75])
```

OUTPUT:

```
Sita        78
Krishna     88
Radha       81
Shivan      99
Parvathi    95
dtype: int64
```

3) Write a program to generate a series of float numbers from 21.0 to 30.0 with an increment of 1.5 each.

CODE:

```
import pandas as pd
import numpy as np
n=np.arange(21.0,30.0,1.5)
s=pd.Series(n)
print(s)
```

OUTPUT:

```
0    21.0
1    22.5
2    24.0
3    25.5
4    27.0
5    28.5
dtype: float64
```

4) Write a program to generate series using dictionary to represent month number and month names.

CODE:

```
import pandas as pd
month={'January':1,'February':2,'March':3,'April':4,'May':5,'June':6,'July':7
    ,'August':8,'September':9,'October':10,'November':11,'December':12}
s1=pd.Series(month)
print(s1)
```

OUTPUT:

```
January      1
February     2
March        3
April        4
May          5
June         6
July         7
August       8
September    9
October     10
November    11
December    12
dtype: int64
```

5) Write a program to generate a series of 5 elements of multiples of 7 starting with 35 with index multiplied by 3.

CODE:

```
import pandas as pd
import numpy as np
n=np.arange(35,70,7)
s1=pd.Series(n,index=n*3)
print(s1)
```

OUTPUT:

```
105    35
126    42
147    49
168    56
189    63
dtype: int64
```

6) Write a program to generate series of marks of 10 students. If marks are < 33, give grace of 5 marks and print new list of marks.

CODE:

```
import pandas as pd
marks=[]
for i in range(1,11):
    print("Roll number: ",i)
    a=int(input("Enter the marks:"))
    marks.append(a)
s=pd.Series(marks)
print(s)
s[s<33]=s+5
print("New list:")
print(s)
```

OUTPUT:

```
Roll number:  1
Enter the marks:50
Roll number:  2
Enter the marks:45
Roll number:  3
Enter the marks:28
Roll number:  4
Enter the marks:17
Roll number:  5
Enter the marks:45
Roll number:  6
Enter the marks:48
Roll number:  7
Enter the marks:31
Roll number:  8
Enter the marks:19
Roll number:  9
Enter the marks:27
Roll number:  10
Enter the marks:31
```

```
0    50
1    45
2    28
3    17
4    45
5    48
6    31
7    19
8    27
9    31
dtype: int64
New list:
0    50
1    45
2    33
3    22
4    45
5    48
6    36
7    24
8    32
9    36
dtype: int64
```

7) Write a program to create a DataFrame of quarterly sales where each row contains item category, item name and expenditure. Group the rows by category and print total expenditure per category.

CODE:

```
import pandas as pd
car={'Category':['Hyundai','Maruti','Mahindra','Audi','Mercedes','Tata','tata'
    ,'Tata'],'Name':['i10','Ertiga','Thar','Q5','G-class','Harrier','Nexon'
    ,'Tiago'],'Expenditure':[1234255,678954,435637,243578,78943,999756,57689
    ,9832567]}
df=pd.DataFrame(car)
print(df)
dfc=df.groupby('Category')
print("DataFrame grouped by category is:")
print(dfc.groups)
print(dfc[['Category','Expenditure']].sum())
```

OUTPUT:

```
   Category     Name  Expenditure
0   Hyundai      i10      1234255
1    Maruti   Ertiga       678954
2  Mahindra     Thar       435637
3      Audi       Q5       243578
4  Mercedes  G-class        78943
5      Tata  Harrier       999756
6      tata    Nexon        57689
7      Tata    Tiago      9832567
DataFrame grouped by category is:
{'Audi': [3], 'Hyundai': [0], 'Mahindra': [2], 'Maruti': [1], 'Mercedes': [4], 'Tata'
    : [5, 7], 'tata': [6]}
          Category  Expenditure
Category
Audi          Audi       243578
Hyundai    Hyundai      1234255
Mahindra  Mahindra       435637
Maruti      Maruti       678954
Mercedes  Mercedes        78943
Tata      TataTata     10832323
tata          tata        57689
```

8) Write a program to create a DataFrame for student data– Name, Percentage, PhoneNumber, Hobbies. Display row labels, column labels and data type of each column and dimensions.

CODE:

```
import pandas as pd
stu=[('Ram',56,12345,'Reading'),('Lakshman',67,543267,'Drawing'),('Sita',89
    ,789234)]
abc=pd.DataFrame(stu,columns=['Name','Marks','Phone','Hobby'],index=[1,2,3])
print(abc)
print("Columns names are:",abc.columns.values)
print("Row names/indices are:",abc.index.values)
print("Dimensions are:",abc.ndim)
```

OUTPUT:

```
       Name  Marks   Phone    Hobby
1       Ram     56   12345  Reading
2  Lakshman     67  543267  Drawing
3      Sita     89  789234     None
Columns names are: ['Name' 'Marks' 'Phone' 'Hobby']
Row names/indices are: [1 2 3]
Dimensions are: 2
```

9) Write a program to accept data from user dynamically and change it in a DataFrame.

CODE:

```
import pandas as pd
import numpy as np
exam={'Name':['A','B','C','D','E','F','G','H','I','J'],'Percentage'
    :[89,np.nan,78,45,67,78,89,76,np.nan,88],'Grade':['A2',np.nan,'B1'
    ,'C2','B2','A2','B1','B1',np.nan,'A1']}
labels=['P','Q','R','S','T','U','V','W','X','Y']
df=pd.DataFrame(exam,index=labels)
print('Original DataFrames is:\n',df)
ind=input('Enter index of row for which you want to change the data:')
per=float(input('Enter Percentage:'))
gr=input('Enter Grade:')
df.loc[ind,'Percentage']=per
df.loc[ind,'Grade']=gr
print('Updated DataFrame is:\n',df)
```

OUTPUT:

```
Original DataFrames is:
   Name  Percentage Grade
P    A        89.0    A2
Q    B         NaN   NaN
R    C        78.0    B1
S    D        45.0    C2
T    E        67.0    B2
U    F        78.0    A2
V    G        89.0    B1
W    H        76.0    B1
X    I         NaN   NaN
Y    J        88.0    A1
Enter index of row for which you want to change the data:X
Enter Percentage:97
Enter Grade:A1
Updated DataFrame is:
   Name  Percentage Grade
P    A        89.0    A2
Q    B         NaN   NaN
R    C        78.0    B1
S    D        45.0    C2
T    E        67.0    B2
U    F        78.0    A2
V    G        89.0    B1
W    H        76.0    B1
X    I        97.0    A1
Y    J        88.0    A1
```

10) Write a program to create a DataFrame and print the top and bottom 3 values of DataFrame.

CODE:

```
import pandas as pd
import numpy as np
exam={'Name':['A','B','C','D','E','F','G','H','I','J'],'Percentage':[89,np.nan
    ,78,45,67,78,89,76,np.nan,88],'Grade':['A2',np.nan,'B1','C2','B2','A2','B1'
    ,'B1',np.nan,'A1']}
labels=['P','Q','R','S','T','U','V','W','X','Y']
df=pd.DataFrame(exam,index=labels)
print('Original DataFrames is:\n',df)
print('Top 3 rows:\n',df.head(3))
print('Bottom 3 rows :\n',df.tail(3))
```

OUTPUT:

```
Original DataFrames is:
   Name  Percentage Grade
P    A        89.0    A2
Q    B         NaN   NaN
R    C        78.0    B1
S    D        45.0    C2
T    E        67.0    B2
U    F        78.0    A2
V    G        89.0    B1
W    H        76.0    B1
X    I         NaN   NaN
Y    J        88.0    A1
Top 3 rows:
   Name  Percentage Grade
P    A        89.0    A2
Q    B         NaN   NaN
R    C        78.0    B1
Bottom 3 rows :
   Name  Percentage Grade
W    H        76.0    B1
X    I         NaN   NaN
Y    J        88.0    A1
```

11) Write a program to create 2 DataFrames. Display their sum and difference.

CODE:

```
import pandas as pd
a={'A':[10,20,30],'B':[40,50,60],'C':[70,80,90]}
b={'A':[56,67,78],'B':[43,54,65,],'C':[22,55,88]}
c=pd.DataFrame(a)
d=pd.DataFrame(b)
print('Sum:\n',c+d)
print('Difference:\n',c-d)
```

OUTPUT:

```
Sum:
      A    B    C
0   66   83   92
1   87  104  135
2  108  125  178
Difference:
     A  B   C
0 -46 -3  48
1 -47 -4  25
2 -48 -5   2
```

12) Write a program to create the DataFrame below. Rename Zone C and Zone D as 'Central' and 'South'.

CODE:

```
import pandas as pd
a={'Target': [56000, 70000, 75000, 60000], 'Sales': [58000, 68000, 78000,
    61000]}
df=pd.DataFrame(a, index=['Zone A', 'Zone B', 'Zone C', 'Zone D'])
print(df)
df=df.rename(index={'Zone C': 'Central', 'Zone D': 'South'}, columns={'Target'
    :'Targeted', 'Sales': 'Achieved'})
print('Renamed dataframe')
print(df)
```

OUTPUT:

```
        Target  Sales
Zone A   56000  58000
Zone B   70000  68000
Zone C   75000  78000
Zone D   60000  61000
Renamed dataframe
         Targeted  Achieved
Zone A      56000     58000
Zone B      70000     68000
Central     75000     78000
South       60000     61000
```

13) Write a program to create a DataFrame of Doctor with columns (Name, Department, Experience). Add a new column Bonus with values as twice as the experience and delete the third row with index 3.

CODE:

```
import pandas as pd
doctor={'Name': ['Dr. Pal', 'Dr.Jadhav', 'Dr. Joshi', 'Dr.Kulkarni'],
    'Department': ['Orthopedic', 'Dermatoligist', 'ENT', 'Cardiology'],
    'Experience': [4,7,7,9]}
a1=pd.DataFrame(doctor, index=[1,2,3,4])
print('Original Dataframe: \n',a1)
a1['Bonus']=a1['Experience']*2
print("Dataframe after adding bonus \n",a1)
a1=a1.drop(3)
print('Data After deleting third row \n',a1)
```

OUTPUT:

```
Original Dataframe:
           Name     Department  Experience
1      Dr. Pal     Orthopedic           4
2    Dr.Jadhav  Dermatoligist           7
3    Dr. Joshi            ENT           7
4  Dr.Kulkarni     Cardiology           9
Dataframe after adding bonus
           Name     Department  Experience  Bonus
1      Dr. Pal     Orthopedic           4      8
2    Dr.Jadhav  Dermatoligist           7     14
3    Dr. Joshi            ENT           7     14
4  Dr.Kulkarni     Cardiology           9     18
Data After deleting third row
           Name     Department  Experience  Bonus
1      Dr. Pal     Orthopedic           4      8
2    Dr.Jadhav  Dermatoligist           7     14
4  Dr.Kulkarni     Cardiology           9     18
```

14) Write a program to create a DataFrame. Print original DataFrame. Filter and print duplicate data in data frame.

CODE:

```
import pandas as pd
abc={'Name': ['Ram', 'Krishna', 'Madhav', 'Ram'], 'Physics': [34,88,23,34],
'Chemistry': [88,88,23,88],
'Maths': [23,88,23,23]}
df=pd.DataFrame(abc)
print('Original DataFrame is: \n',df)
print('Duplicate Data: ')
print(df [df.duplicated (keep=False)])
```

OUTPUT:

```
Original DataFrame is:
       Name  Physics  Chemistry  Maths
0       Ram       34         88     23
1   Krishna       88         88     88
2    Madhav       23         23     23
3       Ram       34         88     23
Duplicate Data:
   Name  Physics  Chemistry  Maths
0   Ram       34         88     23
3   Ram       34         88     23
```

15) Create a DataFrame and extract data row-wise.

CODE:

```
import pandas as pd
abc={'Physics':{'UnitTest1':34,'Term1':88,'UnitTest2':23,'Term2':98},
                'Chemistry':{'UnitTest1':34,'Term1':88,'UnitTest2':23,'Term2'
                    :98},
                'Maths':{'UnitTest1':34,'Term1':88,'UnitTest2':23,'Term2':98}}
df=pd.DataFrame(abc)
print('Original DataFrame is:\n',df)
for i,j in df.iterrows():
    print('Row Indexing:',i)
    print('Containing:',j)
```

OUTPUT:

```
Original DataFrame is:
            Physics  Chemistry  Maths
UnitTest1       34         34     34
Term1           88         88     88
UnitTest2       23         23     23
Term2           98         98     98
Row Indexing: UnitTest1
Containing: Physics      34
Chemistry    34
Maths        34
Name: UnitTest1, dtype: int64
Row Indexing: Term1
Containing: Physics      88
Chemistry    88
Maths        88
Name: Term1, dtype: int64
Row Indexing: UnitTest2
Containing: Physics      23
Chemistry    23
Maths        23
Name: UnitTest2, dtype: int64
Row Indexing: Term2
Containing: Physics      98
Chemistry    98
Maths        98
Name: Term2, dtype: int64
```

16) Write a program to create an employee database that consists of employee ID, Name, Department of 10 employees as a CSV file and create a DataFrame from it.

CSV FILE:

Employee	Name	Department
8369	Smith	Clerk
8370	Anya	Salesman
8371	Seth	Salesman
8372	Madhavar	Manager
8373	Anoop	Clerk
8374	Bina	Manager
8375	Shivansh	Clerk
8376	Scott	Analyst
8377	Amir	President
8378	Kuldeep	Clerk
8379	Jatin	Analyst

CODE:

```
import pandas as pd
df = pd.read_csv("Employee.csv")
print(df)
```

OUTPUT:

```
     Unnamed: 0 Unnamed: 1  Unnamed: 2
0   Employee_ID       Name  Department
1          8369      Smith       Clerk
2          8370       Anya    Salesman
3          8371       Seth    Salesman
4          8372   Madhavan     Manager
5          8373      Anoop       Clerk
6          8374       Bina     Manager
7          8375   Shivansh       Clerk
8          8376      Scott     Analyst
9          8377       Amir   President
10         8378    Kuldeep       Clerk
11         8379      Jatin     Analyst
```

17) Write a program to create a DataFrame of marks of 3 subjects of 5 students and write it in CSV file.

CODE:

```
import pandas as pd
stu={'Roll_No.':[1,2,3,4,5],'Name':['Ram','Lakshman','Sita','Krishna'
    ,'Radha'],
     'Maths':[80,85,87,97,65],'Physics':[76,87,56,45,92],
     'Chemistry':[90,87,67,98,88]}
a=pd.DataFrame(stu)
print(a)
a.to_csv("C:\\Users\\student\\Downloads\\Python Practical1\\Student
    .csv")
```

OUTPUT:

```
   Roll_No.      Name  Maths  Physics  Chemistry
0         1       Ram     80       76         90
1         2  Lakshman     85       87         87
2         3      Sita     87       56         67
3         4   Krishna     97       45         98
4         5     Radha     65       92         88
```

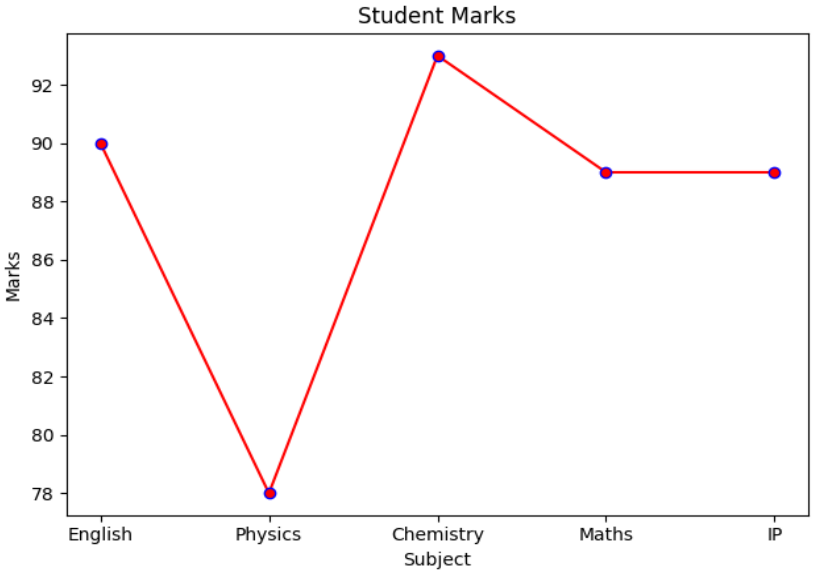
	Roll_No.	Name	Maths	Physics	Chemistry
0	1	Ram	80	76	90
1	2	Lakshman	85	87	87
2	3	Sita	87	56	67
3	4	Krishna	97	45	98
4	5	Radha	65	92	88

18) Write a program to plot a line chart of a student's marks that analyses his data.

CODE:

```
import matplotlib.pyplot as pt
Subject=['English','Physics','Chemistry','Maths','IP']
Marks=[90,78,93,89,89]
pt.plot(Subject,Marks,'r',marker='o',markeredgecolor='blue')
pt.xlabel('Subject')
pt.ylabel('Marks')
pt.title('Student Marks')
pt.show()
```

OUTPUT:

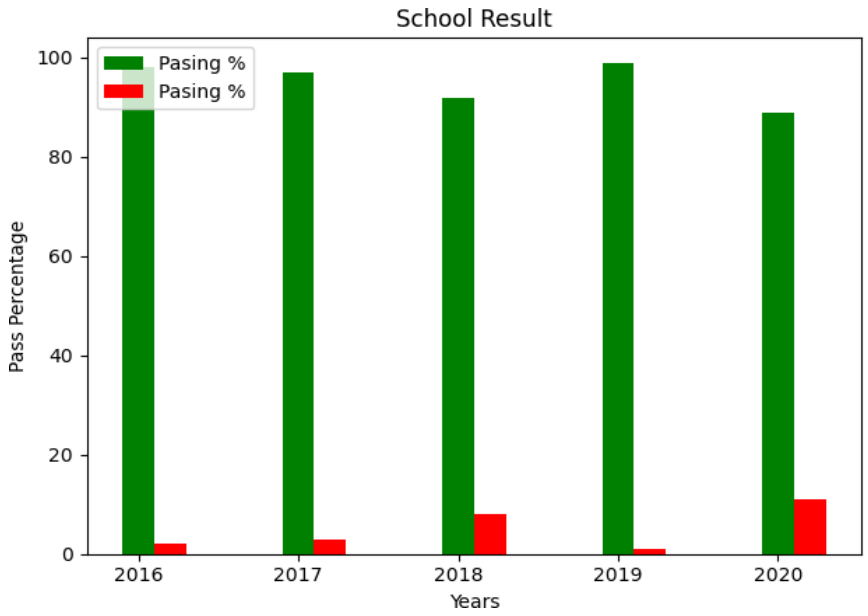


19) Write a program to plot a barchart in python to display result of school for five consecutive years.

CODE:

```
import matplotlib.pyplot as pt
import numpy as np
years=['2016','2017','2018','2019','2020']
pass_pert=[98,97,92,99,89]
fail_pert=[2,3,8,1,11]
X=np.arange(len(years))
pt.bar(years,pass_pert,color='g',width=0.2,label='Pasing %')
pt.bar(X+0.2,fail_pert,color='r',width=0.2,label='Pasing %')
pt.xlabel('Years')
pt.ylabel('Pass Percentage')
pt.title('School Result')
pt.legend(loc='upper left')
pt.show()
```

OUTPUT:



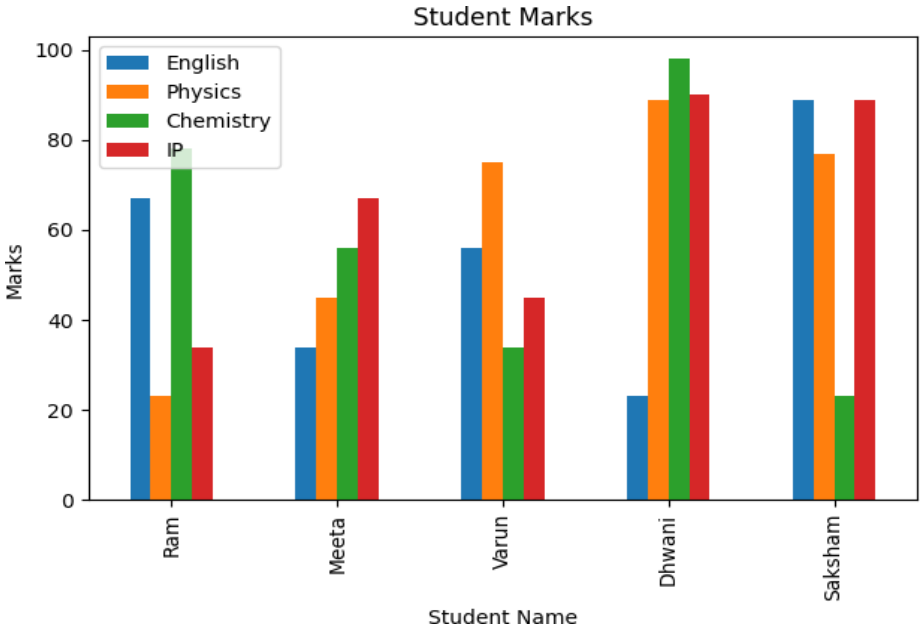
20) Create a bar graph for the DataFrame given below.

CODE:

```
import matplotlib.pyplot as pt
import pandas as pd
marks={'English':[67,34,56,23,89],'Physics':[23,45,75,89,77]
    ,'Chemistry':[78,56,34,98,23],'IP':[34,67,45,90,89]}
a=pd.DataFrame(marks,index=['Ram','Meeta','Varun','Dhwani','Saksham'])
print(a)
a.plot(kind='bar')
pt.xlabel('Student Name')
pt.ylabel('Marks')
pt.title('Student Marks')
pt.show()
pt.savefig()
```

OUTPUT:

	English	Physics	Chemistry	IP
Ram	67	23	78	34
Meeta	34	45	56	67
Varun	56	75	34	45
Dhwani	23	89	98	90
Saksham	89	77	23	89

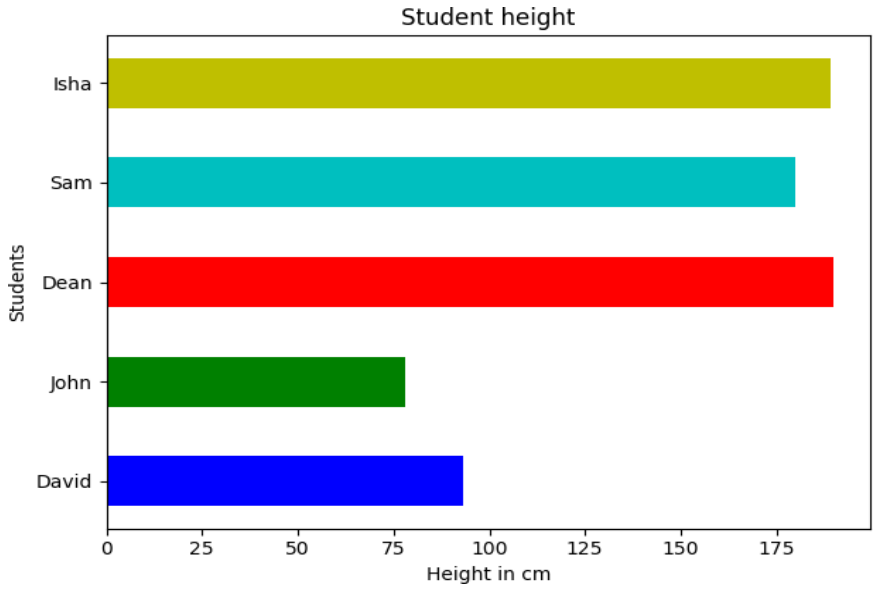


21) Write a program to plot a horizontal bar chart from the heights of 5 students using proper title, label, legends.

CODE:

```
import matplotlib.pyplot as pt
students=['David', 'John', 'Dean', 'Sam', 'Isha']
h=[93,78,190,180,189]
pt.barh(students,h,color=['b','g','r','c','y'],height=0.5)
pt.ylabel('Students')
pt.xlabel('Height in cm')
pt.title('Student height')
pt.show()
```

OUTPUT:



22) Write a program to take data from an open source (data.gov.in).

CSV FILE:

India/State/Union Territory	Population 2011	Growth Rate	Population Density
India	1210854977	17.7	368
Andhra Pradesh	49386799	9	308
Arunachal Pradesh	1383727	9	17
Assam	31205576	17.1	398
Bihar	104099452	25.4	1106
Chhattisgarh	25545198	22.6	189
Goa	1458545	8.2	394
Gujarat	60439692	19.3	308
Haryana	25351462	19.9	573
Himachal Pradesh	6864602	12.9	123

CODE:

```
import pandas as pd
df=pd.read_csv('data.csv')
print(df)
```

OUTPUT:

```
   India/State/Union Territory  Population 2011  Growth Rate  Population Density
0                        India       1210854977         17.7                 368
1              Andhra Pradesh         49386799          9.0                 308
2           Arunachal Pradesh          1383727          9.0                  17
3                        Assam         31205576         17.1                 398
4                        Bihar        104099452         25.4                1106
5                 Chhattisgarh         25545198         22.6                 189
6                          Goa          1458545          8.2                 394
7                      Gujarat         60439692         19.3                 308
8                      Haryana         25351462         19.9                 573
9             Himachal Pradesh          6864602         12.9                 123
```

SQL QUERIES

1. **Write a query to create a student table with student_id, name, marks, phno, address.**

```
mysql> create table student(
    -> student_id int primary key,
    -> name varchar(35),
    -> class varchar(20),
    -> dob date,
    -> address varchar(150),
    -> previous_class_percentage decimal(7,4));
Query OK, 0 rows affected (0.15 sec)
```

```
mysql> desc student;
+--------------------------+--------------+------+-----+---------+-------+
| Field                    | Type         | Null | Key | Default | Extra |
+--------------------------+--------------+------+-----+---------+-------+
| student_id               | int          | NO   | PRI | NULL    |       |
| name                     | varchar(35)  | YES  |     | NULL    |       |
| class                    | varchar(20)  | YES  |     | NULL    |       |
| dob                      | date         | YES  |     | NULL    |       |
| address                  | varchar(150) | YES  |     | NULL    |       |
| previous_class_percentage | decimal(7,4) | YES  |     | NULL    |       |
+--------------------------+--------------+------+-----+---------+-------+
6 rows in set (0.05 sec)
```

2. **Write a query to insert records in student table.**

```
mysql> insert into student values(1, 'Karthik', 'XI', '2007-12-31', 'Pradhikaran', 80);
Query OK, 1 row affected (0.22 sec)

mysql> insert into student values(2, 'Krishna', 'XII', '2007-01-07', 'Nigdi',85);
Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(3, 'Rakshita', 'XII', '2007-05-12', 'Pradhikaran',93);
Query OK, 1 row affected (0.02 sec)

mysql> insert into student values(4, 'Kartik', 'XI', '2007-11-17', 'Akrudi',95);
Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(5, 'Areen', 'XII', '2007-08-14', 'Pradhikaran',79);
Query OK, 1 row affected (0.02 sec)

mysql> insert into student values(6, 'Rajas', 'XII', '2007-10-03', 'Chinchwad',85);
Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(7, 'Kripan', 'XI', '2007-07-07', 'Nigdi',90);
Query OK, 1 row affected (0.03 sec)

mysql> insert into student values(8, 'Kale', 'XII', '2007-01-01', 'Akrudi',84);
Query OK, 1 row affected (0.00 sec)

mysql> insert into student values(9, 'Katik', 'XII','2007-02-28', 'Nigdi',87);
Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(10, 'Krishna', 'XI', '2007-04-30', 'Nigdi',91);
Query OK, 1 row affected (0.01 sec)

mysql> insert into student values(11, 'Nobody', 'XII', '2007-05-31', 'Chinchwad',81);
Query OK, 1 row affected (0.01 sec)
```

```
mysql> select*from student;
+------------+----------+-------+------------+-------------+---------------------------+
| student_id | name     | class | dob        | address     | previous_class_percentage |
+------------+----------+-------+------------+-------------+---------------------------+
|          1 | Karthik  | XI    | 2007-12-31 | Pradhikaran |                   80.0000 |
|          2 | Krishna  | XII   | 2007-01-07 | Nigdi       |                   85.0000 |
|          3 | Rakshita | XII   | 2007-05-12 | Pradhikaran |                   93.0000 |
|          4 | Kartik   | XI    | 2007-11-17 | Akrudi      |                   95.0000 |
|          5 | Areen    | XII   | 2007-08-14 | Pradhikaran |                   79.0000 |
|          6 | Rajas    | XII   | 2007-10-03 | Chinchwad   |                   85.0000 |
|          7 | Kripan   | XI    | 2007-07-07 | Nigdi       |                   90.0000 |
|          8 | Kale     | XII   | 2007-01-01 | Akrudi      |                   84.0000 |
|          9 | Katik    | XII   | 2007-02-28 | Nigdi       |                   87.0000 |
|         10 | Krishna  | XI    | 2007-04-30 | Nigdi       |                   91.0000 |
|         11 | Nobody   | XII   | 2007-05-31 | Chinchwad   |                   81.0000 |
+------------+----------+-------+------------+-------------+---------------------------+
11 rows in set (0.09 sec)
```

3. **Write a query to display details of students with marks > 80.**

```
mysql> select name, previous_class_percentage from student where previous_class_percentage>80;
+----------+---------------------------+
| name     | previous_class_percentage |
+----------+---------------------------+
| Krishna  |                   85.0000 |
| Rakshita |                   93.0000 |
| Kartik   |                   95.0000 |
| Rajas    |                   85.0000 |
| Kripan   |                   90.0000 |
| Kale     |                   84.0000 |
| Katik    |                   87.0000 |
| Krishna  |                   91.0000 |
| Nobody   |                   81.0000 |
+----------+---------------------------+
9 rows in set (0.16 sec)
```

4. **Write a query to delete details of a student from above table.**

```
mysql> delete from student where student_id=11;
Query OK, 1 row affected (0.19 sec)

mysql> select*from student;
+------------+----------+-------+------------+-------------+---------------------------+
| student_id | name     | class | dob        | address     | previous_class_percentage |
+------------+----------+-------+------------+-------------+---------------------------+
|          1 | Karthik  | XI    | 2007-12-31 | Pradhikaran |                   80.0000 |
|          2 | Krishna  | XII   | 2007-01-07 | Nigdi       |                   85.0000 |
|          3 | Rakshita | XII   | 2007-05-12 | Pradhikaran |                   93.0000 |
|          4 | Kartik   | XI    | 2007-11-17 | Akrudi      |                   95.0000 |
|          5 | Areen    | XII   | 2007-08-14 | Pradhikaran |                   79.0000 |
|          6 | Rajas    | XII   | 2007-10-03 | Chinchwad   |                   85.0000 |
|          7 | Kripan   | XI    | 2007-07-07 | Nigdi       |                   90.0000 |
|          8 | Kale     | XII   | 2007-01-01 | Akrudi      |                   84.0000 |
|          9 | Katik    | XII   | 2007-02-28 | Nigdi       |                   87.0000 |
|         10 | Krishna  | XI    | 2007-04-30 | Nigdi       |                   91.0000 |
+------------+----------+-------+------------+-------------+---------------------------+
10 rows in set (0.02 sec)
```

5. **Write a query to display all names in lower case.**

```
mysql> select lcase(name) from student;
+--------------+
| lcase(name) |
+--------------+
| karthik     |
| krishna     |
| rakshita    |
| kartik      |
| areen       |
| rajas       |
| kripan      |
| kale        |
| katik       |
| krishna     |
+--------------+
10 rows in set (0.12 sec)
```

6. **Write a query to display complete table after rounding off data to 1 decimal place.**

```
mysql> select student_id, name, class, dob, address, round(previous_class_percentage, 2) from student;
+------------+----------+-------+------------+-------------+-----------------------------------+
| student_id | name     | class | dob        | address     | round(previous_class_percentage, 2) |
+------------+----------+-------+------------+-------------+-----------------------------------+
|          1 | Karthik  | XI    | 2007-12-31 | Pradhikaran |                             80.00 |
|          2 | Krishna  | XII   | 2007-01-07 | Nigdi       |                             85.00 |
|          3 | Rakshita | XII   | 2007-05-12 | Pradhikaran |                             93.00 |
|          4 | Kartik   | XI    | 2007-11-17 | Akrudi      |                             95.00 |
|          5 | Areen    | XII   | 2007-08-14 | Pradhikaran |                             79.00 |
|          6 | Rajas    | XII   | 2007-10-03 | Chinchwad   |                             85.00 |
|          7 | Kripan   | XI    | 2007-07-07 | Nigdi       |                             90.00 |
|          8 | Kale     | XII   | 2007-01-01 | Akrudi      |                             84.00 |
|          9 | Katik    | XII   | 2007-02-28 | Nigdi       |                             87.00 |
|         10 | Krishna  | XI    | 2007-04-30 | Nigdi       |                             91.00 |
+------------+----------+-------+------------+-------------+-----------------------------------+
10 rows in set (0.12 sec)
```

7. **Write a query to display occurrence of string ‘a’ in names.**

```
mysql> select student_id, instr(name, 'ta'), class, dob, address, previous_class_percentage from student;
+------------+------------------+-------+------------+-------------+--------------------------+
| student_id | instr(name, 'ta') | class | dob        | address     | previous_class_percentage |
+------------+------------------+-------+------------+-------------+--------------------------+
|          1 |                0 | XI    | 2007-12-31 | Pradhikaran |                  80.0000 |
|          2 |                0 | XII   | 2007-01-07 | Nigdi       |                  85.0000 |
|          3 |                7 | XII   | 2007-05-12 | Pradhikaran |                  93.0000 |
|          4 |                0 | XI    | 2007-11-17 | Akrudi      |                  95.0000 |
|          5 |                0 | XII   | 2007-08-14 | Pradhikaran |                  79.0000 |
|          6 |                0 | XII   | 2007-10-03 | Chinchwad   |                  85.0000 |
|          7 |                0 | XI    | 2007-07-07 | Nigdi       |                  90.0000 |
|          8 |                0 | XII   | 2007-01-01 | Akrudi      |                  84.0000 |
|          9 |                0 | XII   | 2007-02-28 | Nigdi       |                  87.0000 |
|         10 |                0 | XI    | 2007-04-30 | Nigdi       |                  91.0000 |
+------------+------------------+-------+------------+-------------+--------------------------+
10 rows in set (0.04 sec)
```

8. **Write a query to display weekday of date of birth of all students.**

```
mysql> select dayname(dob) from student;
+---------------+
| dayname(dob)  |
+---------------+
| Monday        |
| Sunday        |
| Saturday      |
| Saturday      |
| Tuesday       |
| Wednesday     |
| Saturday      |
| Monday        |
| Wednesday     |
| Monday        |
+---------------+
```

9. **Write a query to display count of names of students starting from ‘M’.**

```
mysql> select count(*) from student where name like'M%';
+----------+
| count(*) |
+----------+
|        0 |
+----------+
1 row in set (0.07 sec)
```

10. **Write a query to find min, max, sum and average of marks in student table.**

```
mysql> select min(previous_class_percentage) as 'Min',
    -> max(previous_class_percentage) as 'Max',
    -> sum(previous_class_percentage) as 'Sum',
    -> avg(previous_class_percentage) as 'Avg'
    -> from student;
+---------+---------+---------+-------------+
| Min     | Max     | Sum     | Avg         |
+---------+---------+---------+-------------+
| 79.0000 | 95.0000 | 869.0000 | 86.90000000 |
+---------+---------+---------+-------------+
1 row in set (0.08 sec)
```

11. **Write SQL Query to print student_id and marks in the descending order of marks.**

```
mysql> select student_id,
    -> name,
    -> previous_class_percentage
    -> from student order by previous_class_percentage desc;
+------------+----------+---------------------------+
| student_id | name     | previous_class_percentage |
+------------+----------+---------------------------+
|          4 | Kartik   |                   95.0000 |
|          3 | Rakshita |                   93.0000 |
|         10 | Krishna  |                   91.0000 |
|          7 | Kripan   |                   90.0000 |
|          9 | Katik    |                   87.0000 |
|          2 | Krishna  |                   85.0000 |
|          6 | Rajas    |                   85.0000 |
|          8 | Kale     |                   84.0000 |
|          1 | Karthik  |                   80.0000 |
|          5 | Areen    |                   79.0000 |
+------------+----------+---------------------------+
10 rows in set (0.02 sec)
```

12. **Write a query to find total number of students residing in a particular area using group by.**

```
mysql> select address,count(*) from student group by address;
+-------------+----------+
| address     | count(*) |
+-------------+----------+
| Pradhikaran |        3 |
| Nigdi       |        4 |
| Akrudi      |        2 |
| Chinchwad   |        1 |
+-------------+----------+
4 rows in set (0.17 sec)
```

13. **Write a query to :**

A. **Create a table marks that stores marks of students for 3 subjects physics, chemistry, maths,**
student names.

B. **Join the two tables (student and marks) and print the complete data.**

```
mysql> create table marks (Student_ID int, Physics int, Chemistry int, Maths int);
Query OK, 0 rows affected (0.04 sec)

mysql> insert into marks values
    -> (1,90,90,90),
    -> (2,83,84,85),
    -> (3,95,96,97),
    -> (4,89,88,87),
    -> (5,83,84,85),
    -> (6,87,88,89),
    -> (7,80,81,82),
    -> (8,82,84,86),
    -> (9,88,92,94),
    -> (10,95,96,97);
Query OK, 10 rows affected (0.09 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> select*from marks;
+------------+---------+-----------+-------+
| Student_ID | Physics | Chemistry | Maths |
+------------+---------+-----------+-------+
|          1 |      90 |        90 |    90 |
|          2 |      83 |        84 |    85 |
|          3 |      95 |        96 |    97 |
|          4 |      89 |        88 |    87 |
|          5 |      83 |        84 |    85 |
|          6 |      87 |        88 |    89 |
|          7 |      80 |        81 |    82 |
|          8 |      82 |        84 |    86 |
|          9 |      88 |        92 |    94 |
|         10 |      95 |        96 |    97 |
+------------+---------+-----------+-------+
10 rows in set (0.07 sec)
```

14. **Write a query to print name, address and physics, chemistry, maths marks.**

```
mysql> select name, address, physics, chemistry, maths from student join marks on student.student_id=marks.student_id;
+----------+-------------+---------+-----------+-------+
| name     | address     | physics | chemistry | maths |
+----------+-------------+---------+-----------+-------+
| Karthik  | Pradhikaran |      90 |        90 |    90 |
| Krishna  | Nigdi       |      83 |        84 |    85 |
| Rakshita | Pradhikaran |      95 |        96 |    97 |
| Kartik   | Akrudi      |      89 |        88 |    87 |
| Areen    | Pradhikaran |      83 |        84 |    85 |
| Rajas    | Chinchwad   |      87 |        88 |    89 |
| Kripan   | Nigdi       |      80 |        81 |    82 |
| Kale     | Akrudi      |      82 |        84 |    86 |
| Katik    | Nigdi       |      88 |        92 |    94 |
| Krishna  | Nigdi       |      95 |        96 |    97 |
+----------+-------------+---------+-----------+-------+
10 rows in set (0.11 sec)
```

15. **Write query to print name and total of physics, chemistry & maths marks.**

```
mysql> select name, physics+chemistry+maths as 'total' from student join marks on student.student_id = marks.student_id;
+----------+-------+
| name     | total |
+----------+-------+
| Karthik  |   270 |
| Krishna  |   252 |
| Rakshita |   288 |
| Kartik   |   264 |
| Areen    |   252 |
| Rajas    |   264 |
| Kripan   |   243 |
| Kale     |   252 |
| Katik    |   274 |
| Krishna  |   288 |
+----------+-------+
10 rows in set (0.18 sec)
```