School or CBSE Logo

CENTRAL BOARD OF SECONDARY EDUCATION

School Name
Address

A PRACTICAL RECORD FILE IS SUBMITTED TO DEPARTMENT OF INFORMATICS PRACTICES FOR THE PARTIAL FULLFILLMENT OF AISSCE EXAMINATION SESSION - _____

SUBMITTED BY: [NAME OF STUDENT]

HOD(COMPUTER):[NAME OF SUBJECT TEACHER]

CLASS: [CLASS]

ROLL NO: [XXXXXXX]

ACKNOWLEDGEMENT

I wish to express my deep sense of gratitude and indebtedness to our learned teacher ***TEACHER'S NAME*** , PGT COMPUTER SCIENCE, [SCHOOL NAME] for his invaluable help, advice and guidance in the preparation of this project.

I am also greatly indebted to our principal **[Name of principal]** and school authorities for providing me with the facilities and requisite laboratory conditions for making this practical file.

I also extend my thanks to a number of teachers ,my classmates and friends who helped me to complete this practical file successfully.

[Name of Student]

CERTIFICATE

This is to certify that **[Name of Student]** , student of Class XII, **[NAME OF SCHOOL]** has completed the **PRACTICAL FILE** during the academic year [SESSION] towards partial fulfillment of credit for the Informatics Practices practical evaluation of SSCE-20__ and submitted satisfactory report, as compiled in the following pages, under my supervision.

Total number of practical certified are : 20.

Internal Examiner
Signature

External Examiner
Signature

Date:

School Seal

Principal
Signature



No	Practical	Date	Signature
1	**Data handling using Pandas**		

1	Find the sum of those values which are ending with 3 or 5.		
2	Create a series of 10 numbers starting with 41 and with the increment of 3. Now add 7 all odd values and subtract 3 in even values. Reprint the updated series.		
3	Create a series of 10 numbers. Change the value of all the elements those values are multiples of 4.		
4	Create a series and print the top 3 elements using the head function.		
5	Create a series and print the last 3 elements using the tail function.		
6	Create a series with these numbers: 21, 51, 71, 31, 12. Exchange all these values of series by shifting each of them one to one position before and by shifting the first value to last position.		
7	Create a dataframe named as students using a list of names of 5 students		
8	Create a dataframe players using a list of names and scores of the previous three matches. (Using Nested list)		
9	Create a dataframe **countries** using a dictionary which stored country name, capitals and populations of the country.		
10	Iterate dataframe created in question no. 8 by its rows.		
11	Print scores of previous two matches along with their names using iterrows function. (Use dataframe created in question 8)		
12	Make a total of score from the dataframe players and display their rank according their scores.		
13	Print the batsman name along with runs scored in Test and T20 using column names and dot notation.		
No	**Practical**	**Date**	**Signature**
14	Display the Batsman name along with runs scored in ODI using loc.		
15	Display the batsman details who scored ● More than 2000 in ODI		

<table>
<tr><td></td><td>

- Less than 2500 in Test
- More than 1500 in T20

</td><td></td><td></td></tr>
<tr><td>2</td><td colspan="3">Part 2 Data Visualization</td></tr>
<tr><td>16</td><td>Plot following data on line chart and follow the given instructions:
<table>
<tr><td>Day</td><td>Monday</td><td>Tuesday</td><td>Wednesday</td><td>Thursday</td><td>Friday</td></tr>
<tr><td>Income</td><td>510</td><td>350</td><td>475</td><td>580</td><td>600</td></tr>
</table>

- Write a title for the chart "The Weekly Income Report".
- Write the appropriate titles of both the axes.
- Write code to Display legends.
- Display red color for the line.
- Use the line style - dashed
- Display diamond style markers on data points

</td><td></td><td></td></tr>
<tr><td>17</td><td>Consider the following data of a medical store and plot the data on the line chart and Customize the chart as you wish:
<table>
<tr><td>Month</td><td>Masks</td><td>Sanitizer</td><td>Hand wash</td></tr>
<tr><td>March</td><td>1500</td><td>4400</td><td>6500</td></tr>
<tr><td>April</td><td>3500</td><td>4500</td><td>5000</td></tr>
<tr><td>May</td><td>6500</td><td>5500</td><td>5800</td></tr>
<tr><td>June</td><td>6700</td><td>6000</td><td>6300</td></tr>
<tr><td>July</td><td>6000</td><td>5600</td><td>6200</td></tr>
<tr><td>August</td><td>6800</td><td>6300</td><td>4500</td></tr>
</table>
</td><td></td><td></td></tr>
<tr><td>18</td><td>Use above data and subplot sanitizer data and handwash data.</td><td></td><td></td></tr>
<tr><td>19</td><td>Display following bowling figures through bar chart:
<table>
<tr><td>Overs</td><td>Runs</td></tr>
<tr><td>1</td><td>6</td></tr>
<tr><td>2</td><td>18</td></tr>
<tr><td>3</td><td>10</td></tr>
<tr><td>4</td><td>5</td></tr>
</table>
</td><td></td><td></td></tr>
<tr><td>No</td><td>Practical</td><td>Date</td><td>Signature</td></tr>
<tr><td>3</td><td>Part 3 MySQL queries</td><td></td><td></td></tr>
<tr><td></td><td>Consider the given table and write given queries :
1) To join product and company and display in the tabular form like - <pname> manufactured by <company></td><td></td><td></td></tr>
</table>

	2) Convert all product name into capital 3) Display the cube of products quantity more than or 100 4) Divide the price by 3 and display the result with 1 fraction digit for the price of more than 40,000. 5) Display pname (last four letters only), qty, price with 2 decimal points and company for price in between 30000 to 80000. 6) Display maximum price of products 7) Display the total quantities of all products 8) Display the average price of LED TV and Apple products 9) Find the difference between maximum price and minimum price from the table. 10) Display unique Products from the table. 11) Count the unique company from products. 12) Display the product number, product name and company in the descending order of their price. 13) Display product minimum price for each company. 14) Display product number and product names in their ascending order of names. 15) Display maximum price of products manufactured by apple.		

Part 1: Data Handling Using Pandas

1. **Create a series of these numbers:**

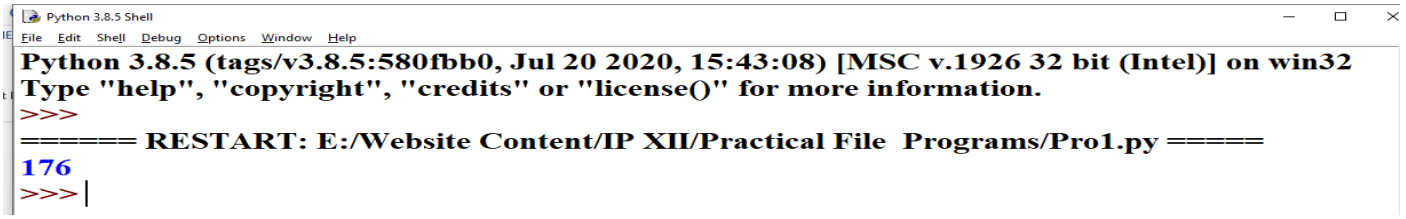
   33,55,65,29,19,23.

   Find the sum of those values which are ending with 3 or 5.

   Code:



Output:



2. **Create a series of 10 numbers starting with 41 and with the increment of 3. Now add 7 all odd values and subtract 3 in even values. Reprint the updated series.**

Code:

```
import pandas as pd
ser=pd.Series(range(41,71,3))
for i in range(0,ser.size):
    if ser[i]%2==0:
        ser[i]=ser[i]-3
    elif ser[i]%2!=0:
        ser[i]=ser[i]+7
print(ser)
```

Output:

```
0    48
1    41
2    54
3    47
4    60
5    53
6    66
7    59
8    72
9    65
dtype: int64
```

3. **Create a series of 10 numbers. Change the value of all the elements those values are multiples of 4.**

Code:

Pro1.py - E:/Website Content/IP XII/Practical File Programs/Pro1.py (3.8.5)

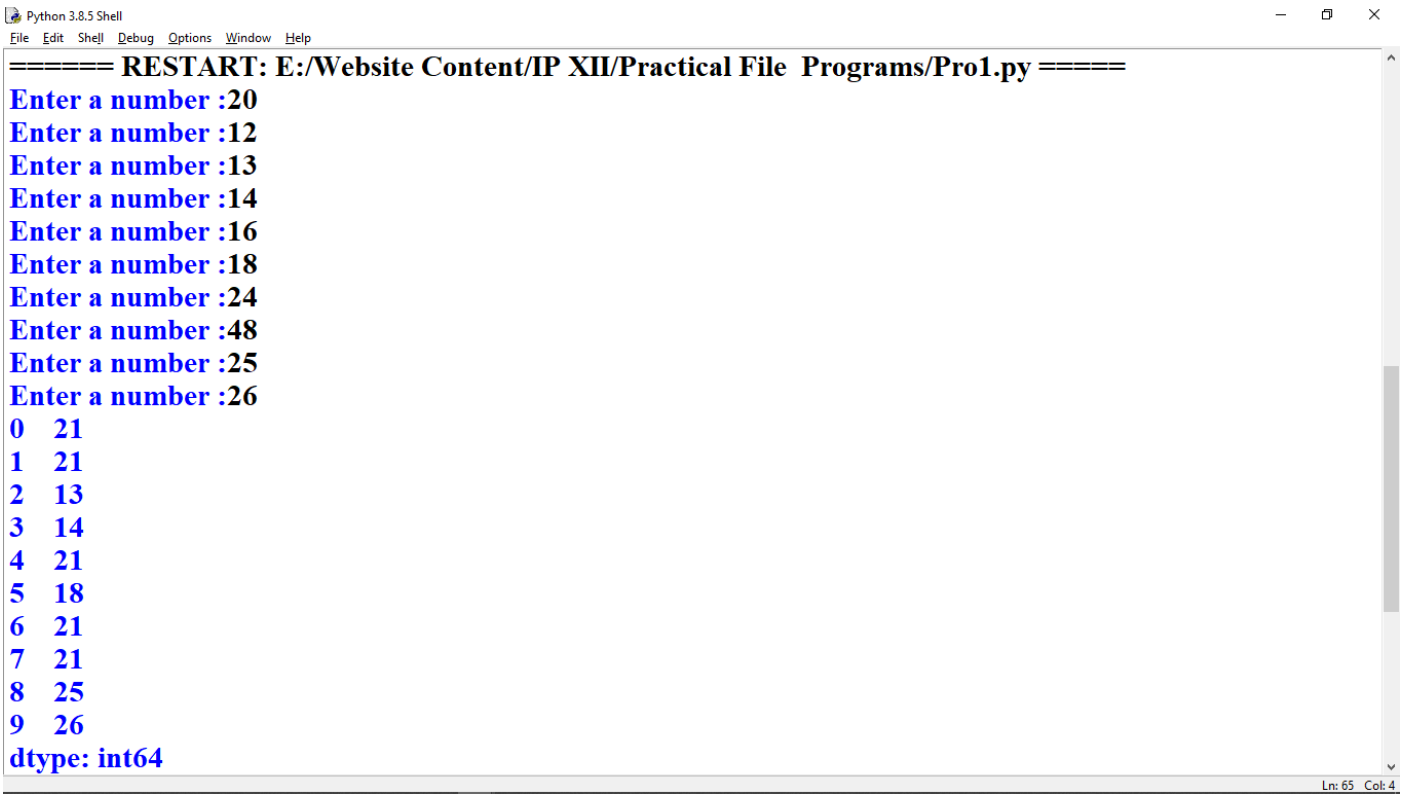
File Edit Format Run Options Window Help

```
import pandas as pd
numbers = []

# Generating a series
for i in range(1,11):
    val = int(input("Enter a number :"))
    numbers.append(val)
ser = pd.Series(numbers)

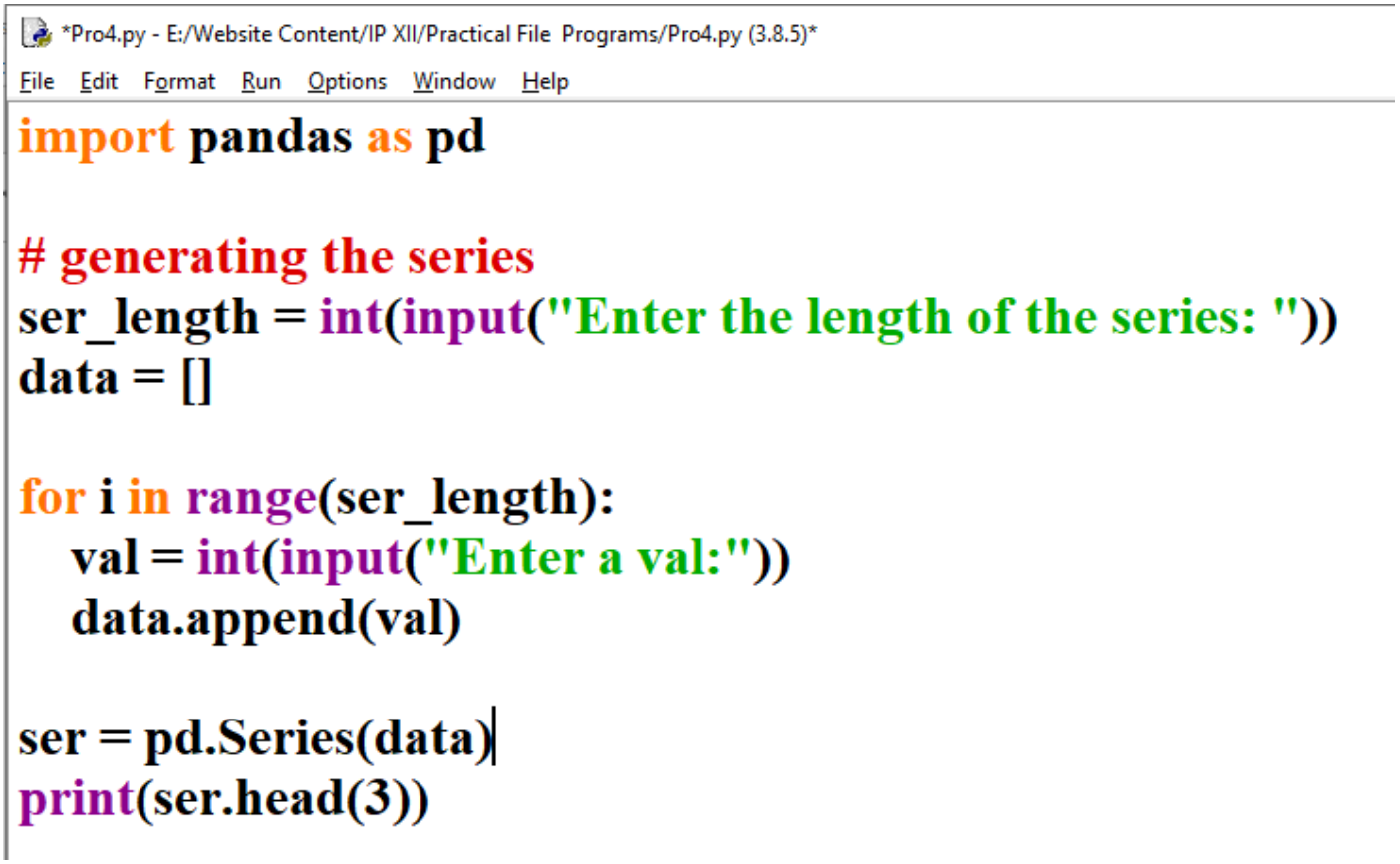
# Changing the values of multiple of four and assigning them a value 21
ser[ser % 4 == 0] = 21
print(ser)
```

Output:



4. **Create a series and print the top 3 elements using the head function.**

Code:



Output:

```
>>>
========== RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro4.py =========
Enter the length of the series: 5
Enter a val:11
Enter a val:20
Enter a val:55
Enter a val:23
Enter a val:65
0   11
1   20
2   55
dtype: int64
```

5. **Create a series and print the bottom 3 elements using the tail function.**

Code:

Pro5.py - E:/Website Content/IP XII/Practical File Programs/Pro5.py (3.8.5)

File Edit Format Run Options Window Help

```
import pandas as pd

# generating the series
ser_length = int(input("Enter the length of the series: "))
data = []

for i in range(ser_length):
    val = int(input("Enter a val:"))
    data.append(val)


ser = pd.Series(data)

#Printing 3 elements
print(ser.tail(3))
```

```
>>>
========== RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro5.py =========
Enter the length of the series: 5
Enter a val:21
Enter a val:25
Enter a val:63
Enter a val:85
Enter a val:74
2   63
3   85
4   74
dtype: int64
>>>
```

6. **Create a series with these numbers: 21, 51, 71, 31, 12. Exchange all these values of series by shifting each of them one to one position before and by shifting the first value to last position..**

Code:

Pro6.py - E:/Website Content/IP XII/Practical File Programs/Pro6.py (3.8.5)

File Edit Format Run Options Window Help

```
import pandas as pd
import numpy as np
s = pd.Series([21,51,71,31,12])
print(pd.Series(np.roll(s.values, -1), index=s.index))
```

Output:

```
>>>
========== RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro6.py =========
0   51
1   71
2   31
3   12
4   21
dtype: int64
```

7. **Create a dataframe named as students using a list of names of 5 students.**

Code:

Pro7.py - E:/Website Content/IP XII/Practical File Programs/Pro7.py (3.8.5)

File Edit Format Run Options Window Help

```
import pandas as pd
students = ["Ram","Aman","Akash","Ramesh","Virat"]
students = pd.DataFrame(students,columns=["Name"])
print(students)
```

Output:

Python 3.8.5 Shell

File Edit Shell Debug Options Window Help

```
Python 3.8.5 (tags/v3.8.5:580fbb0, Jul 20 2020, 15:43:08) [MSC v.1926 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======= RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro7.py ======
    Name
0    Ram
1   Aman
2  Akash
3 Ramesh
4  Virat
```

8. **Create a dataframe players using a list of names and scores of the previous three matches. (Using Nested list)**

Code:

```
Pro8.py - E:/Website Content/IP XII/Practical File  Programs/Pro8.py (3.8.5)
File  Edit  Format  Run  Options  Window  Help
import pandas as pd
data = [["Virat",55,66,31],["Rohit",88,66,43],["Dhoni",99,85,68]]
players = pd.DataFrame(data, columns = ["Name","M1","M2","M3"])
print(players)
```

Output:

```
>>>
====== RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro8.py
   Name  M1  M2  M3
0  Virat  55  66  31
1  Rohit  88  66  43
2  Dhoni  99  85  68
```

9. **Create a dataframe** countries **using a dictionary which stored country name, capitals and populations of the country.**

Code:

```
Pro9.py - E:\Website Content\IP XII\Practical File  Programs\Pro9.py (3.8.5)
File  Edit  Format  Run  Options  Window  Help
import pandas as pd
country_data = {"Country Name":["India","Canada","Australia"],
            "Capital": ["New Delhi","Ottawa","Canberra"],
            "Population" : ["136 Cr","10 Cr","50 Cr"]}
countries = pd.DataFrame(country_data)
print(countries)
```

Output:

```
>>>
====== RESTART: E:\Website Content\IP XII\Practical File  Programs\Pro9.py =====
  Country Name    Capital Population
0       India  New Delhi     136 Cr
1      Canada     Ottawa      10 Cr
2   Australia   Canberra      50 Cr
```

10. **Iterate dataframe created in question no. 8 by its rows.**

Code:

Pro10.py - E:/Website Content/IP XII/Practical File Programs/Pro10.py (3.8.5)

File Edit Format Run Options Window Help

```
import pandas as pd

#Creating Dataframe
data = [["Virat",55,66,31],["Rohit",88,66,43],["Samson",99,101,68]]
players = pd.DataFrame(data, columns = ["Name","Match-1","Match-2","Match-3"])

#Iterating dataframe using iterrows
for index, row in players.iterrows():
    print(index, row.values)
```

Output:

```
>>>
===== RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro10.py =====
0 ['Virat' 55 66 31]
1 ['Rohit' 88 66 43]
2 ['Samson' 99 101 68]
>>>
```

11. **Print scores of previous two matches along with their names using iterrows function. (Use dataframe created in question 8)**

Code:

Pro11.py - E:/Website Content/IP XII/Practical File Programs/Pro11.py (3.8.5)

File Edit Format Run Options Window Help

```
import pandas as pd

data = [["Virat",55,66,31],["Rohit",88,66,43],["Samson",99,101,68]]
players = pd.DataFrame(data, columns = ["Name","Match-1","Match-2","Match-3"])

for index, row in players.iterrows():
    print(index, row["Name"],row["Match-2"],row["Match-3"])
```

Output:

```
===== RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro11.py =====
0 Virat 66 31
1 Rohit 66 43
2 Samson 101 68
```

12. **Make a total of score from the dataframe players and display their rank according the their scores.**

Code:

Pro12.py - E:/Website Content/IP XII/Practical File Programs/Pro12.py (3.8.5)

File Edit Format Run Options Window Help

```
import pandas as pd

data = [["Virat",55,66,31],["Rohit",88,66,43],["Dhoni",99,53,68],["Pant",77,55,21]]
players = pd.DataFrame(data, columns = ["Name","Match-1","Match-2","Match-3"])

players['Total_score'] = players['Match-1'] + players['Match-2'] + players["Match-3"]
players['Rank'] = players['Total_score'].rank(ascending=0)
print(players)
```

Output:

```
>>>
===== RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro12.py =====
   Name  Match-1  Match-2  Match-3  Total_score  Rank
0  Virat      55       66       31          152   4.0
1  Rohit      88       66       43          197   2.0
2  Dhoni      99       53       68          220   1.0
3   Pant      77       55       21          153   3.0
>>>
```

13. **Print the batsman name along with runs scored in Test and T20 using column names and dot notation.**

Code:

Pro13.py - E:/Website Content/IP XII/Practical File Programs/Pro13.py (3.8.5)

File Edit Format Run Options Window Help

```
import pandas as pd

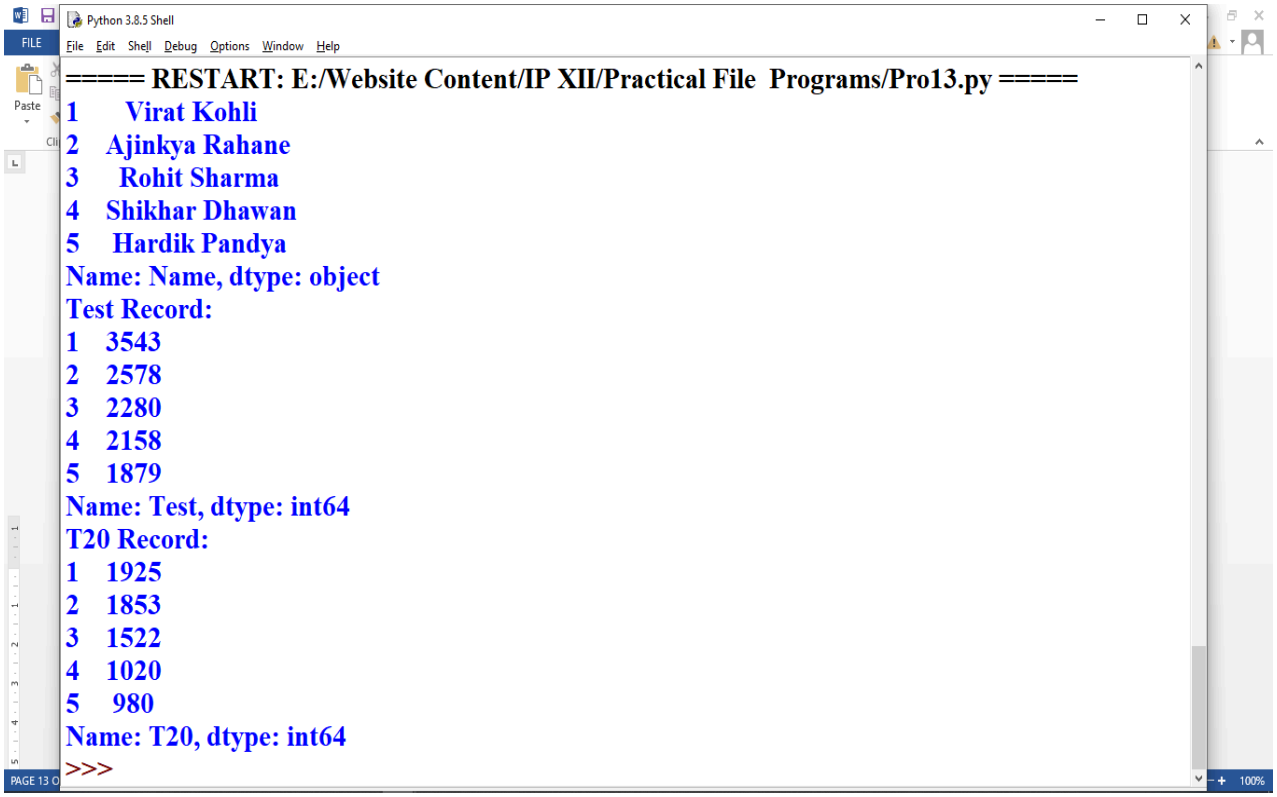
# Creating the Data
player_data = {"Name":["Virat Kohli","Ajinkya Rahane","Rohit Sharma",\
                       "Shikhar Dhawan","Hardik Pandya"],
               "Test" : [3543,2578,2280,2158,1879],
               "ODI" : [2245,2165,2080,1957,1856],
               "T20" : [1925,1853,1522,1020,980]
}

data = pd.DataFrame(player_data)
# The following line is used to start the index from 1
data.index = data.index + 1

# ---------------------
#Printing Names
print(data.Name)
#Test Record
print("Test Record:")
print(data.Test)
#T20 Record
print("T20 Record:")
```

Ln: 5 Col: 23

Output:

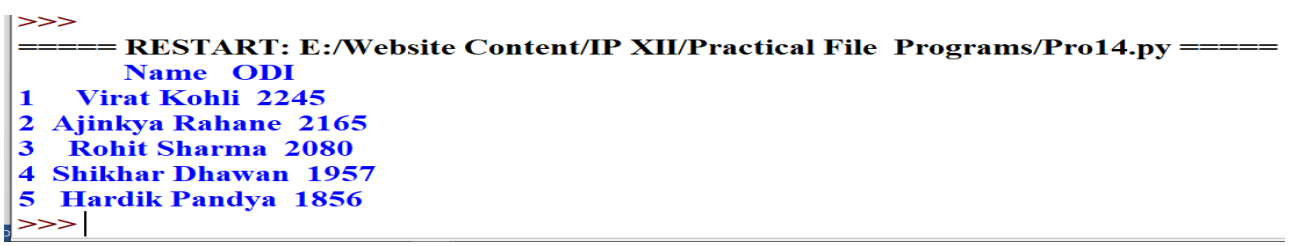


14. **Display the Batsman name along with runs scored in ODI using loc.**

Code:



Output:



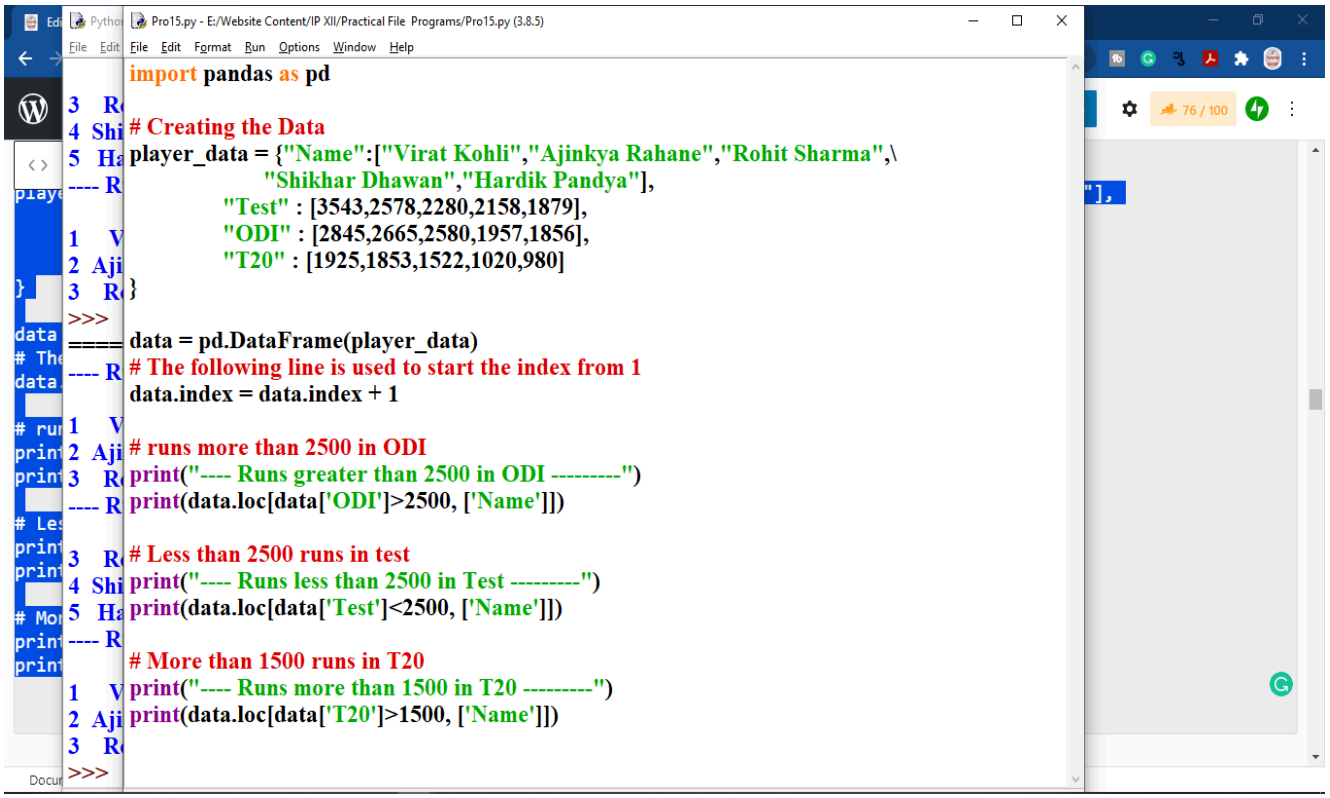
15. **Display the batsman details who scored**

a. More than 2000 in ODI

b. Less than 2500 in Test

c. More than 1500 in T20

Code:



```
import pandas as pd

# Creating the Data
player_data = {"Name":["Virat Kohli","Ajinkya Rahane","Rohit Sharma",\
                       "Shikhar Dhawan","Hardik Pandya"],
               "Test" : [3543,2578,2280,2158,1879],
               "ODI" : [2845,2665,2580,1957,1856],
               "T20" : [1925,1853,1522,1020,980]
}

data = pd.DataFrame(player_data)
# The following line is used to start the index from 1
data.index = data.index + 1

# runs more than 2500 in ODI
print("---- Runs greater than 2500 in ODI ---------")
print(data.loc[data['ODI']>2500, ['Name']])

# Less than 2500 runs in test
print("---- Runs less than 2500 in Test ---------")
print(data.loc[data['Test']<2500, ['Name']])

# More than 1500 runs in T20
print("---- Runs more than 1500 in T20 ---------")
print(data.loc[data['T20']>1500, ['Name']])
```

Output:

```
>>>
===== RESTART: E:/Website Content/IP XII/Practical File  Programs/Pro15.py =====
---- Runs greater than 2500 in ODI ---------
          Name
1    Virat Kohli
2  Ajinkya Rahane
3    Rohit Sharma
---- Runs less than 2500 in Test ---------
          Name
3    Rohit Sharma
4  Shikhar Dhawan
5   Hardik Pandya
---- Runs more than 1500 in T20 ---------
          Name
1    Virat Kohli
2  Ajinkya Rahane
3    Rohit Sharma
```

Part 2: Data Visualization

16. **Plot following data on line chart and follow the given instructions:**

Day	Monday	Tuesday	Wednesday	Thursday	Friday
Income	510	350	475	580	600

a. **Write a title for the chart "The Weekly Income Report".**

b. **Write the appropriate titles of both the axes.**

c. **Write code to Display legends.**

d. **Display red color for the line.**

e. **Use the line style - dashed**

f. **Display diamond style markers on data points**

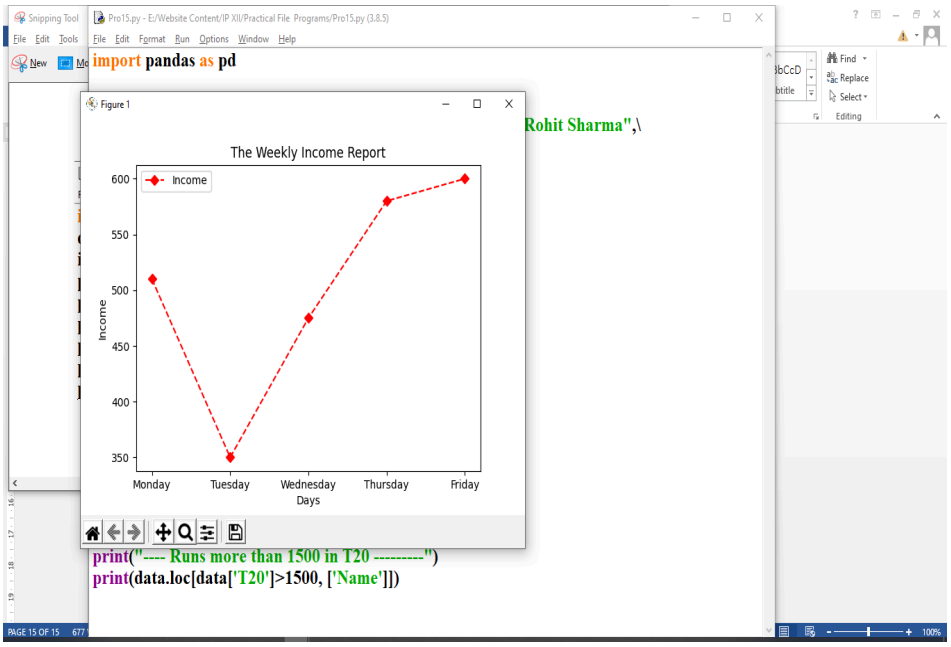
Code:

Pro1.py - E:/Website Content/IP XII/Practical File Programs/Data Visualization/Pro1.py (3.8.5)

File Edit Format Run Options Window Help

```
import matplotlib.pyplot as pp
day =['Monday','Tuesday','Wednesday','Thursday','Friday']
inc = [510,350,475,580,600]
pp.plot(day,inc,label='Income',color='r',linestyle='dashed',marker='D')
pp.title("The Weekly Income Report")
pp.xlabel("Days")
pp.ylabel("Income")
pp.legend()
pp.show()
```

Output:



17. **Consider the following data of a medical store and plot the data on the line chart and Customize the chart as you wish:**

Month	Masks	Sanitizer	Hand wash
March	1500	4400	6500
April	3500	4500	5000
May	6500	5500	5800
June	6700	6000	6300
July	6000	5600	6200
August	6800	6300	4500

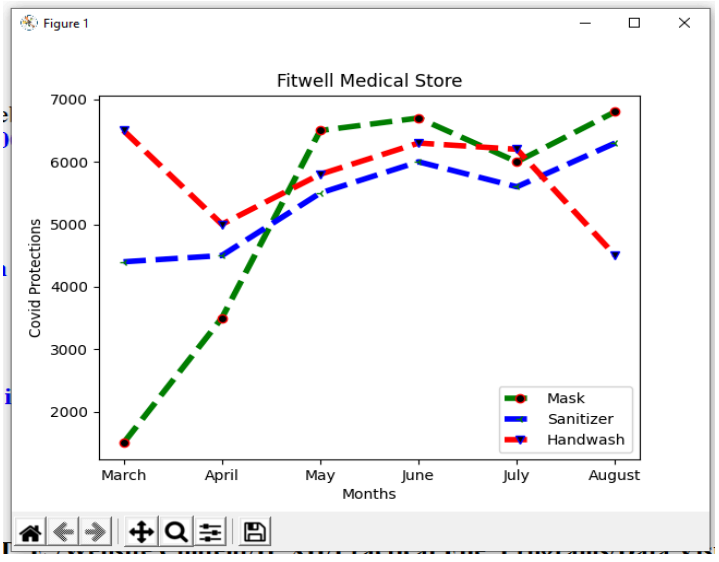
Code:

Pro2.py - E:/Website Content/IP XII/Practical File Programs/Data Visualization/Pro2.py (3.8.5)

File Edit Format Run Options Window Help

```
import matplotlib.pyplot as pp
mon =['March','April','May','June','July','August']
mask= [1500,3500,6500,6700,6000,6800]
san = [4400,4500,5500,6000,5600,6300]
hw = [6500,5000,5800,6300,6200,4500]
pp.plot(mon,mask,label='Mask',color='g',linestyle='dashed', linewidth=4,\
      marker='o', markerfacecolor='k', markeredgecolor='r')
pp.plot(mon,san,label='Sanitizer',color='b',linestyle='dashed', linewidth=4,\
      marker='3', markerfacecolor='k', markeredgecolor='g')
pp.plot(mon,hw,label='Handwash',color='r',linestyle='dashed', linewidth=4,\
      marker='v', markerfacecolor='k', markeredgecolor='b')
pp.title("Fitwell Medical Store")
pp.xlabel("Months")
pp.ylabel("Covid Protections")
pp.legend()
pp.show()
```

Output:



18. **Use above data and subplot sanitizer data and handwash data.**

Code:

Pro3.py - E:/Website Content/IP XII/Practical File Programs/Data Visualization/Pro3.py (3.8.5)

File Edit Format Run Options Window Help

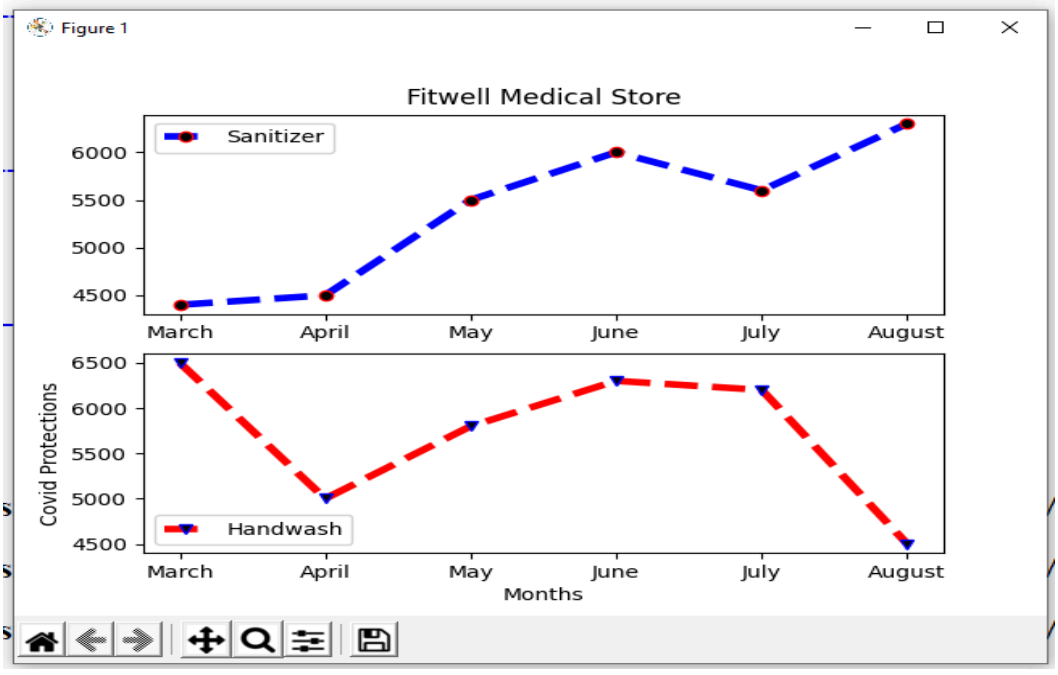
```
import matplotlib.pyplot as pp
mon =['March','April','May','June','July','August']

san = [4400,4500,5500,6000,5600,6300]
hw = [6500,5000,5800,6300,6200,4500]

#subplot1 for Sanitizer
pp.subplot(2,1,1)
pp.plot(mon,san,label='Sanitizer',color='b',linestyle='dashed', linewidth=4,\
      marker='o', markerfacecolor='k', markeredgecolor='r')
pp.title("Fitwell Medical Store")
pp.legend()

#subplot2 for Handwash
pp.subplot(2,1,2)
pp.plot(mon,hw,label='Handwash',color='r',linestyle='dashed', linewidth=4,\
      marker='v', markerfacecolor='k', markeredgecolor='b')
pp.xlabel("Months")
pp.ylabel("Covid Protections")
pp.legend()
pp.show()
```

Output:



19. **Display following bowling figures through bar chart:**

Overs	Runs
1	6
2	18
3	10
4	5

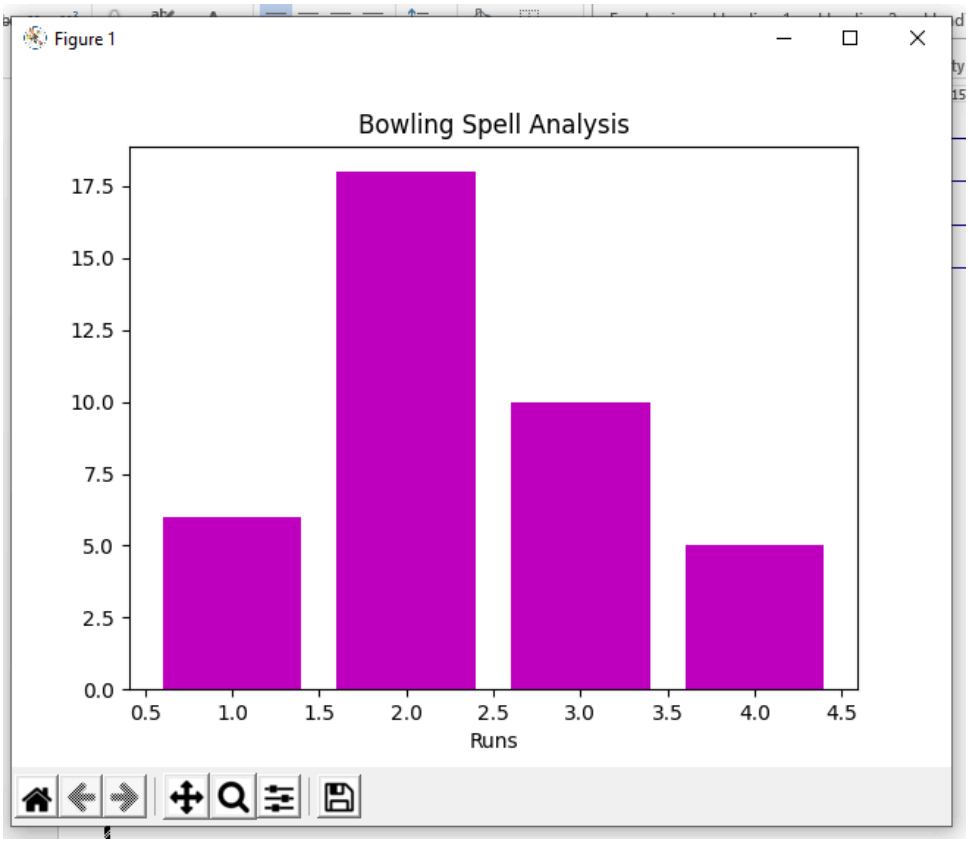
Code:

Pro4.py - E:/Website Content/IP XII/Practical File Programs/Data Visualization/Pro4.py (3.8.5)

File Edit Format Run Options Window Help

```
import matplotlib.pyplot as pp
overs =[1,2,3,4]
runs=[6,18,10,5]
pp.bar(overs,runs,color='m')
pp.xlabel('Overs')
pp.xlabel('Runs')
pp.title('Bowling Spell Analysis')
pp.show()
```

Output



Part 3: Database query using MySQL

20. **Create the following table products and write queries given below:**

Table: Products

Pcode	Pname	Qty	Price	Company
P1001	iPad	120	15000	Apple
P1002	LED TV	100	85000	Sony
P1003	DSLR Camera	10	25000	Philips
P1004	iPhone	50	95000	Apple
P1005	LED TV	20	45000	MI
P1006	Bluetooth Speaker	100	20000	Ahuja

Constraints:

1. Pcode – Primary Key
2. Pname – Not Null

Create table command:

```
mysql> create table products
    -> (pno varchar(5) primary key,
    -> pname varchar(25) not null,
    -> qty int(3),
    -> price int(5),
    -> company varchar(15));
Query OK, 0 rows affected (0.23 sec)
```

Insert record command

```
mysql> insert into products values('P1001','iPad',120,15000,'Apple');
Query OK, 1 row affected (0.08 sec)
```

1. **To join product and company and display in tabular form like - <pname> manufatured by <company>**

```
mysql> select concat(pname,concat(" manufactured by ",company)) from products;

+-------------------------------------------------+
| concat(pname,concat(" manufactured by ",company)) |
+-------------------------------------------------+
| iPad manufactured by Apple                      |
| LED TV manufactured by Sony                     |
| DSLR Camera manufactured by Philips             |
| iPhone manufactured by Apple                    |
| LED TV manufactured by MI                       |
| Bluetooth Speaker manufactured by Ahuja         |
+-------------------------------------------------+
6 rows in set (0.00 sec)
```

2. **Convert all product name into capital**

```
mysql> select ucase(pname) from products;
+--------------------+
| ucase(pname)       |
+--------------------+
| IPAD               |
| LED TV             |
| DSLR CAMERA        |
| IPHONE             |
| LED TV             |
| BLUETOOTH SPEAKER  |
+--------------------+
6 rows in set (0.03 sec)
```

3. **Display the cube of products quantity for more than or 100 in quantity.**

```
mysql> select qty,pow(qty,3) from products where qty>=100;
+------+------------+
| qty  | pow(qty,3) |
+------+------------+
|  120 |    1728000 |
|  100 |    1000000 |
|  100 |    1000000 |
+------+------------+
3 rows in set (0.07 sec)
```

4. **Divide the price by 3 and display the result with 1 fraction digit for price of more than 40,000.**

```
mysql> select round((price/3),1) from products where price>40000;
+--------------------+
| round((price/3),1) |
+--------------------+
|            28333.3 |
|            31666.7 |
|            15000.0 |
+--------------------+
3 rows in set (0.00 sec)
```

5. **Display pname (last four letters only), qty, price with 2 decimal points and company for price in between 30000 to 80000.**

```
mysql> select right(pname,4),qty,round(price,2),company from products where pr
ice between 30000 and 80000;
+---------------+------+---------------+---------+
| right(pname,4) | qty  | round(price,2) | company |
+---------------+------+---------------+---------+
| D TV          |   20 |      45000.00 | MI      |
+---------------+------+---------------+---------+
1 row in set (0.01 sec)
```

6. **Display maximum price of products**

```
mysql> select max(price) from products;
+--------------+
| max(price)   |
+--------------+
|        95000 |
+--------------+
1 row in set (0.02 sec)
```

7. **Display the total quantities of all products**

```
mysql> select sum(qty) from products;
+-----------+
| sum(qty) |
+-----------+
|       400 |
+-----------+
1 row in set (0.08 sec)
```

8. **Display the average price of LED TV and Apple products**

```
mysql> select avg(price) from products where pname='LED TV' or company='Apple'
;
+------------+
| avg(price) |
+------------+
| 60000.0000 |
+------------+
1 row in set (0.00 sec)
```

9. **Find the difference between maximum price and minimum price from the table.**

```
mysql> select max(price)-min(price) from products;
+-----------------------+
| max(price)-min(price) |
+-----------------------+
|                 80000 |
+-----------------------+
1 row in set (0.04 sec)
```

10. **Display unique Products from the table.**

```
mysql> select distinct(pname) from products;
+--------------------+
| pname              |
+--------------------+
| iPad               |
| LED TV             |
| DSLR Camera        |
| iPhone             |
| Bluetooth Speaker  |
+--------------------+
5 rows in set (0.02 sec)
```

11. **Count the unique company from products.**

```
mysql> select count(distinct(company)) from products;
+--------------------------+
| count(distinct(company)) |
+--------------------------+
|                        5 |
+--------------------------+
1 row in set (0.03 sec)
```

12. **Display the product number, product name and company in the descending order of their price.**

```
mysql> select pno,pname,company,price from products order by price desc;
+-------+-------------------+---------+-------+
| pno   | pname             | company | price |
+-------+-------------------+---------+-------+
| P1004 | iPhone            | Apple   | 95000 |
| P1002 | LED TV            | Sony    | 85000 |
| P1005 | LED TV            | MI      | 45000 |
| P1003 | DSLR Camera       | Philips | 25000 |
| P1006 | Bluetooth Speaker | Ahuja   | 20000 |
| P1001 | iPad              | Apple   | 15000 |
+-------+-------------------+---------+-------+
6 rows in set (0.02 sec)
```

13. **Display product minimum price for each company.**

```
mysql> select company,min(price) from products group by company;
+---------+------------+
| company | min(price) |
+---------+------------+
| Ahuja   |      20000 |
| Apple   |      15000 |
| MI      |      45000 |
| Philips |      25000 |
| Sony    |      85000 |
+---------+------------+
5 rows in set (0.02 sec)
```

14. **Display product number and product names in their ascending order of names.**

```
mysql> select pno,pname from products order by pname;
+-------+--------------------+
| pno   | pname              |
+-------+--------------------+
| P1006 | Bluetooth Speaker  |
| P1003 | DSLR Camera        |
| P1001 | iPad               |
| P1004 | iPhone             |
| P1002 | LED TV             |
| P1005 | LED TV             |
+-------+--------------------+
6 rows in set (0.00 sec)
```

15. **Display maximum price of products manufactured by apple.**

```
mysql> select company,max(price) from products group by company having company
='Apple';
+---------+------------+
| company | max(price) |
+---------+------------+
| Apple   |      95000 |
+---------+------------+
1 row in set (0.03 sec)
```