Informatics Practical File

Arnav Mangla

CLASS: 12-C

AMITY INTERNATIONAL SCHOOL
CLASS XII: 2021-22, INFORMATICS PRACTICES (065)
PRACTICAL LIST

PYTHON REVIEW

1. Write a program to read two lists, L1 and L2 from user. Now merge L1 and L2 and print “mostly odd” if merged list contains more odd numbers than even numbers else it should print “mostly even”

2. Write a menu driven program which takes a string as input and as per user choice perform the respective operations.

   Sample Output

   Menu

   1. To count the no. of uppercase and lowercase alphabets.
   2. To check if the string is palindromic
   3. To check repeating characters (yes/no)
   4. To count special characters (any non-alphanumeric character is special)

   Enter choice: 1
   Enter the string: ______
   Uppercase alphabets: ___ Lowercase alphabets: ___
   ______is/is not Palindrome
   Characters are/are not repeating
   Special characters: ____

3. Write a python program to input ‘n’ names and phone numbers to store it in a dictionary with names as keys paired with their phone numbers as values. Your code should now print the following:
   a. The Phone Number of the person whose name is entered by the user.
   b. The Name of the person whose phone is entered by the user.

4. Write a program to read a sentence from the user and create a dictionary with each word as a key and its frequency (no of occurrences) of that word as value. For example, if sentence read is
   “It is true that the joy of giving is true joy’
   Then dictionary should be
   {“It”:1, “is”:2, “true”:2, “that”:1,“the”:1, “joy”:2,”of ”:1,”giving”:1}

PYPLOT

5. Write a Python programming to create a bar chart of gold medal achievements of five most successful countries in 2016 Summer Olympics.
   Sample data: United States, 46 Great
   Britain,27 China,26 Russia,19
   Germany,17

6. Write a Python program to create a bar plot of scores with different color bars for men and women.
   Sample Data:
   Scores (men) = (22, 30, 35, 35, 26)
   Scores (women) = (25, 32, 30, 35, 29)

7. Write a Python programming to display a horizontal bar chart and a pie chart of the popularity of programming Languages.

   Sample data:
   Programming languages: Java, Python, PHP, JavaScript, C#, C++
   Popularity: 22.2, 17.6, 8.8, 8, 7.7,6.7

8. Write a Python program to draw a bar plot comparing two subject marks of Mathematics and Science. Use marks of 10 students.

   Sample Data:

   math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]

   science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]

   marks_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]

9. Write a program to compute the histogram of nums against the bins.

   nums: [0.5 0.7 1. 1.2 1.3 2.1]

   bins: [0 1 2 3]

PANDAS

10. Three Series objects stores the marks of 10 students in 3 terms. Roll numbers of students form the index of these Series objects. The 3 Series objects have the same indexes. calculate the total weighted marks obtained by students as per the following formula:
    Final_marks=25% term1 + 25% term2+ 50% term3
    Store the final marks of students in another series object.
    Perform the following operations on the final marks series:
    a) Display the top 3 marks in the series
    b) Display the bottom 3 marks in the series
    c) Display the average marks of students
    d) Display the maximum marks in the series
    e) Display the minimum marks in the series
    f) Display the marks which are greater than 90

11. Series objectsweek1, week2, week3, week4 store the temperatures of days of 4 weeks of a month respectively. Write a script to:
    a) Print the average temperature per week
    b) Store the average temperature of each day for 4 weeks in another series AVG_TEMP.
    c) Print the maximum temperature from the AVG_Temp Series
    d) Print the minimum temperature from the AVG_Temp Series
    e) Print the days where temperature is above 35 degrees.

12. Create a dataframe “temp” from the four series created above -week1, week2, week3 and week4. The indexes should be ‘sun’, ‘mon’ ….’sat’ and columns should be week1, week2, week3 and week4. Write a script to calculate:
    a) Average temperatures for each day of the week i.e., average temperature for Monday, Tuesday etc.
    b) Average temperature per week
    c) Average temperature for whole month

13. Given a data frame DATA. Write code statement to:

	Company	Quantity	Price
Printer	HCL	32	18000
Printer	Lenovo	7	12000
Scanner	HCL	12	14000
Monitor	Asus	46	15000
Scanner	Samsung	23	17000

a) Find all rows with label “Printer”. Extract all columns.
b) List all the peripherals with quantity more than 30.
c) List single true or false to signify if all prices are more than 15000 or not
d) List 2nd, 3rd and 4th rows.
e) List only the columns quantity and price using **loc.**
f) List only columns 1 and 2 (column indexes) using **iloc.**
g) List only rows with labels “Printer” and “Scanner” using **loc.**
h) Write statement to delete rows with labels “Printer” and “monitor”
i) Write statement to delete columns with labels “company” and “price”

14. Write a python program to create a dataframe called “student” containing the following columns:
Rollno, name,Eng, Phy, Chem, Math, IP
Input the rollno, name and marks in 5 subjects. Calculate the total and percentage
All the newly created columns (totalmarks, per, grade) should be inserted in the dataframe. Print the final dataframe.

15. Write a Python program to read a given CSV file having tab delimiter.

Sample Output

```
country_id country_name region_id
AR        Argentina    2
AU        Australia    3
BE        Belgium      1
BR        Brazil       2
CA        Canada       2
```

16. Write a Python program to read specific columns of a given CSV file and print the content of the columns.

Sample Output

```
ID   Department Name
10   Administration
20   Marketing
30   Purchasing
40   Human Resources
50   Shipping
60    IT
```

SQL AND PYTHON

17. Answer the following questions based on the table TRANSPORT.

a) Create the table and insert the following records

Rtno	Area_Covered	Capacity	No_of_ students	Distance	Transporter	Charges
1	VasantKunj	100	100	10	Shivam Tr.	10000.00
2	Rohini	80	80	10	Anand Tr	55000.75
3	Saket	50	50	30	Bhalla Co.	95000.50
4	Yamuna Vihar	120	120	35	Speed Tr	10000.00
5	Saket	100	100	20	Raj Tr.	80000.00
6	JanakPuri	40	40	10	Yadav Co.	60000.50

Write queries for the following:

b) Display RtNo, Area_Covered and charges(without decimals) for all the school buses.

c) Display all the details from above table in ascending order of Area_Covered

d) Remove column Distance.

e) Increase the charges of Yadav Co. by 1000.

f) Display Area_Covered in capitals and first three characters of the attribute **Transporter**.

g) Display the total charges according to each Area_covered.

h) Display the RtNo and Area Covered for the routes where No_of_students is more than the capacity.

i) Find outputs of the following queries:

i) Select Capacity, No_of_students from Transport where charges>60000;

ii) Select RtNo, charges/4 as "Quarterly Charges" from Transport;

iii) Select Transporter from Transport where Area_Covered like 'S%';

18. Make the following two tables ITEM and CUSTOMER and write the commands for the following

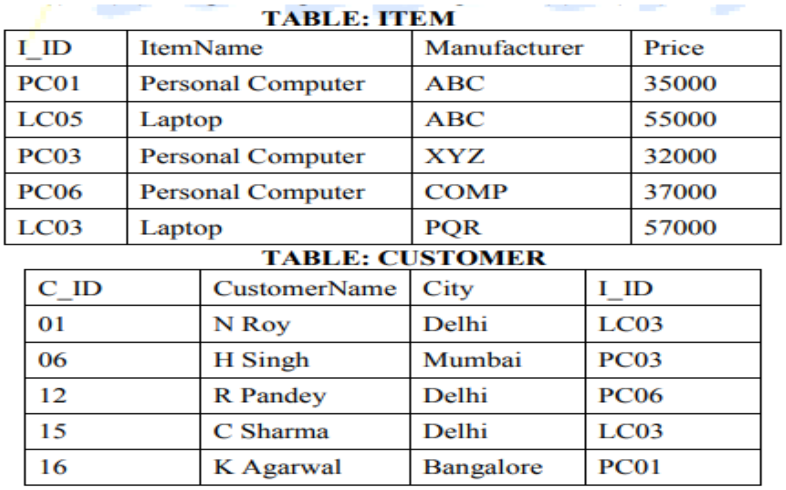


TABLE: ITEM

I_ID	ItemName	Manufacturer	Price
PC01	Personal Computer	ABC	35000
LC05	Laptop	ABC	55000
PC03	Personal Computer	XYZ	32000
PC06	Personal Computer	COMP	37000
LC03	Laptop	PQR	57000

TABLE: CUSTOMER

C_ID	CustomerName	City	I_ID
01	N Roy	Delhi	LC03
06	H Singh	Mumbai	PC03
12	R Pandey	Delhi	PC06
15	C Sharma	Delhi	LC03
16	K Agarwal	Bangalore	PC01

i) To display the details of customers of those items whose Price is in the range of 35000 to 55000 (Both values included).

ii) To display the average price of items based on cities of their customers

(iii) To display the CustomerName, City from table Customer and ItemName from table Item.

iv) To display ItemName and Price from table Item for the customers living in Mumbai

v) To increase the Price of all Items by 1000 in the table Item for all personal computers

vi) To Display details of customers who purchased the costliest item

(vii) Add one more column STOCK as varchar (20) to ITEM table (ii) Display all the records of Student Table

Question – 1

Q : Write a program to read two lists, L1 and L2 from user. Now merge L1 and L2 and print “mostly odd” if merged list contains more odd numbers than even numbers else it should print “mostly even”

Code:

```
def read_list(input_msg):
    l = list(eval(input(input_msg)))
    return l

L1 = read_list("Enter the first list: ")
L2 = read_list("Enter the second list: ")
merged = L1 + L2
even = 0
odd = 0

for element in merged:
    if element%2==0 : even += 1
    else : odd += 1

if even >= odd : print("mostly even")
else : print("mostly odd")
```

Output:

```
Enter the first list: 1, 2, 3,4,5
Enter the second list: 1,2,3
mostly odd
```

Question – 2

Q: Write a menu driven program which takes a string as input and as per user choice perform the respective operations.

Sample Menu

1. To count the no. of uppercase and lowercase alphabets.
2. To check if the string is palindromic
3. To check repeating characters (yes/no)
4. To count special characters (any non alphanumeric character is special)

Code:

```
def menu_choice():
    print("""

        1. To count the no. of uppercase and lowercase alphabets
        2. To check if the string is palindromic
        3. To check repeating characters
        4. To count special characters
        5. Quit""")
    choice = int(input("Enter Choice: "))
    print("\n")
    return choice

def check_case(string):
    upper = 0
    lower = 0
    for letter in string:
        if letter.isalpha():
            if letter.islower() : lower += 1
            else : upper += 1
    print(f"There are {upper} uppercase and {lower} lowercase alphabets in
{string}")
    print("Note : Numbers and special characters are not considered.")

def check_palindrome(string):
      rev_string = string[len(string)::-1]
      if rev_string.lower() == string.lower() : print(f"{string} IS palindromic")
      else : print(f"{string} is NOT palindromic")

def check_repetition(string):
    l = []
    rep = 0
    for char in string.lower():
        if char in l : rep += 1
        else : l.append(char)

    print(f"There are {rep} repeating characters in {string}")

def check_special(string):
    special = 0
    for char in string:
        if not (char.isalnum()) : special += 1
    if special == 1 : print(f"There is 1 special character in {string} ")
    else : print(f"There are {special} special characters in {string} ")
    print("NOTE : Spaces are also counted as special characters")

main = str(input("Enter a string: "))
choice = 0
while choice != 5 :
    choice = menu_choice()
    if choice == 1: check_case(main)
    elif choice == 2: check_palindrome(main)
    elif choice == 3: check_repetition(main)
    elif choice == 4: check_special(main)
    elif choice == 5:
        print("Goodbye!")
        break
    else :
        print("Invalid Choice.\nPlease choose again:")
        continue
```

Output:

```
Enter a string: heLlo Darkness My oLd fRiEnD
```

```
        1. To count the no. of uppercase and lowercase alphabets
        2. To check if the string is palindromic
        3. To check repeating characters
        4. To count special characters
        5. Quit
Enter Choice: 1
There are 7 uppercase and 17 lowercase alphabets in heLlo Darkness My oLd fRiEnD
Note : Numbers and special characters are not considered.

        1. To count the no. of uppercase and lowercase alphabets
        2. To check if the string is palindromic
        3. To check repeating characters
        4. To count special characters
        5. Quit
Enter Choice: 2
heLlo Darkness My oLd fRiEnD is NOT palindromic

        1. To count the no. of uppercase and lowercase alphabets
        2. To check if the string is palindromic
        3. To check repeating characters
        4. To count special characters
        5. Quit
Enter Choice: 3
There are 13 repeating characters in heLlo Darkness My oLd fRiEnD

        1. To count the no. of uppercase and lowercase alphabets
        2. To check if the string is palindromic
        3. To check repeating characters
        4. To count special characters
        5. Quit
Enter Choice: 4
There are 4 special characters in heLlo Darkness My oLd fRiEnD
NOTE : Spaces are also counted as special characters

        1. To count the no. of uppercase and lowercase alphabets
        2. To check if the string is palindromic
        3. To check repeating characters
        4. To count special characters
        5. Quit
Enter Choice: 111
Invalid Choice.
Please choose again:

        1. To count the no. of uppercase and lowercase alphabets
        2. To check if the string is palindromic
        3. To check repeating characters
        4. To count special characters
        5. Quit
Enter Choice: 5
Goodbye!
```

Question – 3

Q: Write a python program to input ‘n’ names and phone numbers to store it in a dictionary with names as keys paired with their phone numbers as values.

Your code should now print the following:

i) The Phone Number of the person whose name is entered by the user.

ii) The Name of the person whose phone is entered by the user.

Code:

```
n = int(input("Enter the number of contacts to be stored: "))
names = []
num = []
for _ in range(n):
    names.append(input("\nEnter NAME of contact: ").lower())
    num.append(input("\nEnter NUMBER of contact: "))
contacts = dict(zip(names, num))
print(contacts)
print("""
    1. Search using Name
    2. Search using Number
    3. Quit
""")
choice = 1
while choice != 3:

    try:
        choice = int(input("Enter choice: "))
    except ValueError:
        print("Invalid Choice. Please choose again\n")
        continue

    if choice not in [1,2,3]:
        print("Invalid Choice. Please choose again\n")
        continue
    elif choice == 1:

        name = input("\nEnter name of contact : ")
        print(name.capitalize() + " : " + contacts.get(name.lower(), "Does not
exist"))

    elif choice == 2:
        num = input("\nEnter number of contact : ")
        for key in contacts:
            if contacts[key] == num:
                print(f"{key.capitalize()} : {num} ")
                break
        else : print("The number does not exist")
    elif choice == 3: break

print("\nGoodbye!")
```

Output:

```
Enter the number of contacts to be stored: 2

Enter NAME of contact: Raj

Enter NUMBER of contact: 123345567

Enter NAME of contact: Advika

Enter NUMBER of contact: 149162536
{'raj': '123345567', 'advika': '149162536'}

    1. Search using Name
    2. Search using Number
    3. Quit

Enter choice: 1

Enter name of contact : Advika
Advika : 149162536
```

```
Enter choice: 2

Enter number of contact : 123345567
Raj : 123345567
Enter choice: 1

Enter name of contact : person_who_does_not_exist
Person_who_does_not_exist : Does not exist
Enter choice: 2

Enter number of contact : number_which_doesn't_exist
The number does not exist
Enter choice: 3

Goodbye!
```

Question – 4

Q: Write a program to read a sentence from the user and create a dictionary with each word as a key and its frequency (no of occurrences) of that word as value.

For example, if sentence is “It is true that the joy of giving is true joy’. Then dictionary should be
{“It”:1, “is”:2,”true”:2.”that”:1,“the”:1, “joy”:2,”of ”:1,”giving”:1}

Code:

```
import json
words = list(map(str, input("Enter a sentence: ").lower().split()))
occ = []
words_distinct = []
for word in words:
    if word not in words_distinct:
        occ.append(1)
        words_distinct.append(word)
    else:
        occ[words_distinct.index(word)] += 1

print(json.dumps(dict(zip(words_distinct, occ)), indent = 3))
```

Output:

```
Enter a sentence: Interestingly enough , the boy lept over the stream overcoming his
fear . The boy knew that , now , he had nothing to fear
{
   "interestingly": 1,
   "enough": 1,
   ",": 3,
   "the": 3,
   "boy": 2,
   "lept": 1,
   "over": 1,
   "stream": 1,
   "overcoming": 1,
   "his": 1,
   "fear": 2,
   ".": 1,
   "knew": 1,
   "that": 1,
   "now": 1,
   "he": 1,
   "had": 1,
   "nothing": 1,
   "to": 1
}
```

Question – 5

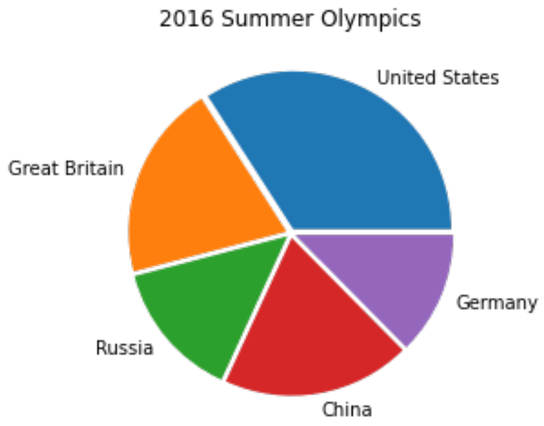
Q : Write a Python programming to create a pie chart of gold medal achievements of five most successful countries in 2016 Summer Olympics.

Sample Data:
United States 46
Great Britain 27
Russia 19
China 26
Germany 17

Code:

```
import matplotlib.pyplot as py

countries = list(map(str, "United States , Great Britain , Russia , China ,
Germany".split(' , ')))
medals = [46, 27, 19, 26, 17]
explode = [0.03]*5
py.title("2016 Summer Olympics")
py.pie(medals, labels=countries, explode=explode)
py.show()
```

Output:



Question – 6

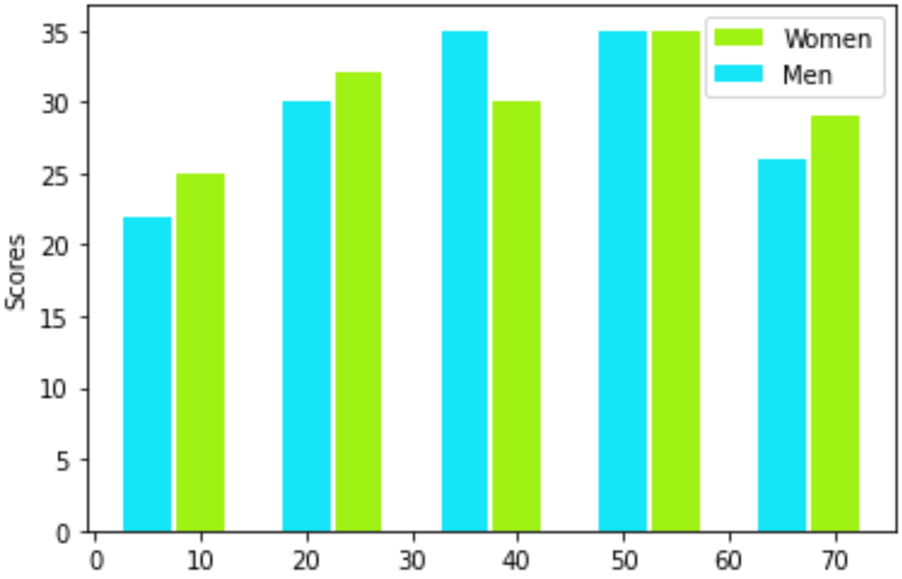
Q : Write a Python program to create a bar plot of scores with different color bars for men and women.

Sample Data:
Scores (men) = (22, 30, 35, 35, 26)
Scores (women) = (25, 32, 30, 35, 29)

Code:

```
from matplotlib import pyplot as py
import numpy as np
men = [22, 30, 35, 35, 26]
women = [25, 32, 30, 35, 29]
xlabels = ["Men", "Women"]
x1 = np.arange(5, 80, 15)
x2= x1+5
py.bar(x2, women,color="#A0F114", label = "Women", width=4.5)
py.bar(x1,men,color="#14E5F7", label = "Men", width=4.5)
py.legend()
py.ylabel("Scores")
py.show()
```

Ouptut:



Question – 7

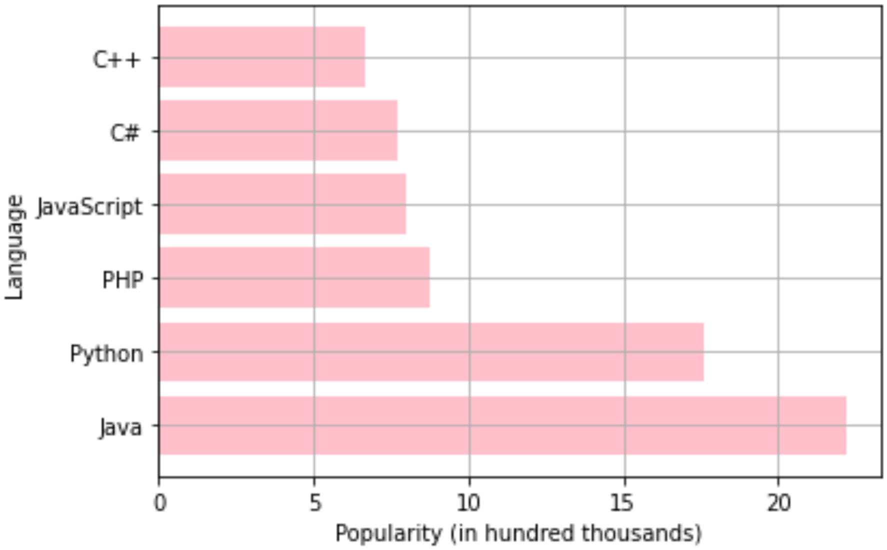
Q : Write a Python programming to display a horizontal bar chart and a pie chart of the popularity of programming Languages.

Sample data:
Programming languages: Java, Python, PHP, JavaScript, C#, C++
Popularity: 22.2, 17.6, 8.8, 8, 7.7,6.7

Code:

```
from matplotlib import pyplot as py
lang = ['Java','Python', 'PHP', 'JavaScript', 'C#', 'C++']
pop = [22.2, 17.6, 8.8, 8, 7.7, 6.7]
py.barh(lang, pop, height=0.8, color="pink")
py.xlabel("Popularity (in hundred thousands)")
py.ylabel("Language")
py.grid()
py.show()
```

Output:



Question – 8

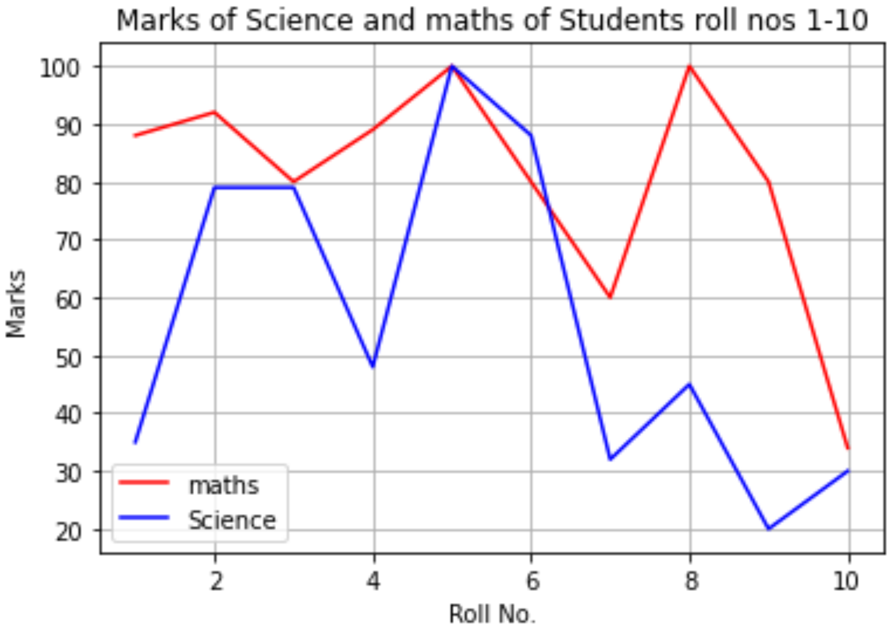
Q : Write a Python program to draw a line chart comparing two subject marks of Mathematics and Science. Use marks of 10 students.

Sample Data:
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]

Code:

```
from matplotlib import pyplot as py
math_marks = [88, 92, 80, 89, 100, 80, 60, 100, 80, 34]
science_marks = [35, 79, 79, 48, 100, 88, 32, 45, 20, 30]
roll_no = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
py.plot(roll_no, math_marks, color="red", label="maths")
py.plot(roll_no, science_marks, color= "blue", label="Science")
py.legend()
py.xlabel("Roll No.")
py.title("Marks of Science andmaths of Students roll nos 1-10")
py.ylabel("Marks")
py.grid()
py.show()
```

Output:



Question – 9

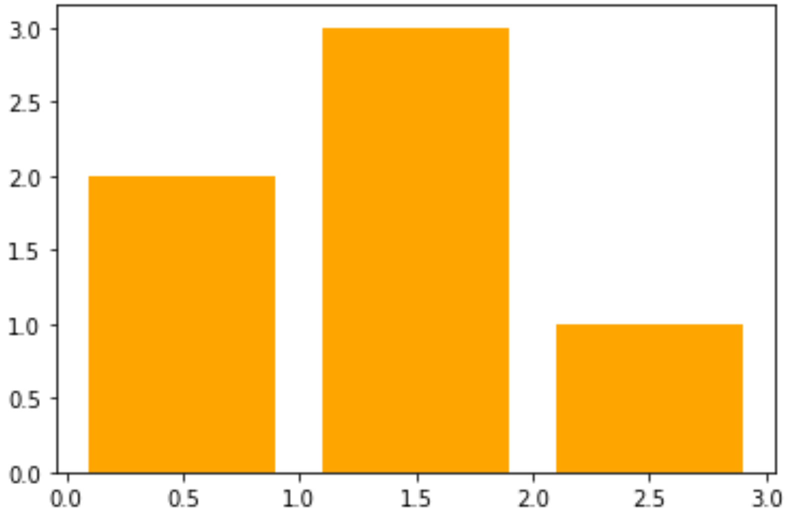
Q : Write a program to compute the histogram of nums against the bins.

Data:
nums: [0.5 0.7 1. 1.2 1.3 2.1]
bins: [0 1 2 3]

Code:

```
from matplotlib import pyplot as py
import numpy as np

nums = [0.5, 0.7, 1.1, 1.2, 1.3, 2.1]
bins = range(0,4,1)
py.hist(nums, bins=bins, rwidth=0.8,color='orange')
py.show()
```

Output:



Question – 10

Q : Three Series objects stores the marks of 10 students in 3 terms. Roll numbers of students form the index of these Series objects.

The 3 Series objects have the same indexes.
Calculate the total weighted marks obtained by students as per the following formula : Final_marks = 25% term1 + 25% term2 + 50%term3
Perform the following operations on the final marks series:
1. Display the top 3 marks in the series
2. Display the bottom 3 marks in the series
3. Display the average marks of students
4. Display the maximum marks in the series
5. Display the minimum marks in the series
6. Display the marks which are greater than 90

Code:

```
import pandas as pd
import random as r
term1 = pd.Series([r.randint(60,100) for _ in range(10)], [x+1 for x in range(10)])
term2 = pd.Series([r.randint(60,100) for _ in range(10)], [x+1 for x in range(10)])
term3 = pd.Series([r.randint(60,100) for _ in range(10)], [x+1 for x in range(10)])
final = pd.Series(term1 * 0.25 + term2 * 0.25 + term3 * 0.5)
print("The final marks are: ")
print(final,"\n")

print("Top 3 marks in the series: ")
print(final.sort_values(ascending = False).head(3),"\n")
print("Bottom 3 marks in the series:")
print(final.sort_values(ascending = False).tail(3), "\n")
print("Average marks of students: ")
print(final.mean())
print("Maximum marks in the series: ")
print(final.max())
print("Minimum marks in the series: ")
print(final.min(),"\n")
print("Marks which are greater than 90: ")
if final[final>90].count == 0: print("No one scored over 90")
else: print(final[final>90])
```

Output:

```
The final marks are:
1     68.50
2     89.50
```

```
3     76.00
4     79.50
5     83.75
6     86.50
7     83.50
8     73.75
9     90.75
10    82.25
dtype: float64

Top 3 marks in the series:
9    90.75
2    89.50
6    86.50
dtype: float64

Bottom 3 marks in the series:
3    76.00
8    73.75
1    68.50
dtype: float64

Average marks of students:
81.4
Maximum marks in the series:
90.75
Minimum marks in the series:
68.5

Marks which are greater than 90:
9    90.75
dtype: float64
```

Question – 11

Q : Series objects week1, week2, week3, week4 store the temperatures of days of 4 weeks of a month respectively.

Write a script to:

1. Print the average temperature per week
2. Store the average temperature of each day for 4 weeks in another series AVG_TEMP.
3. Print the maximum temperature from the AVG_Temp Series
4. Print the minimum temperature from the AVG_Temp Series
5. Print the days where temperature is above 35 degrees.

Code:

```
import pandas as pd
import random as r
index = ["Mon", "Tue", "Wed", "Thu", "Fri", "Sat", "Sun"]
week1 = pd.Series([r.randint(30, 45) for _ in range(7)], index)
week2 = pd.Series([r.randint(30, 45) for _ in range(7)], index)
week3 = pd.Series([r.randint(30, 45) for _ in range(7)], index)
week4 = pd.Series([r.randint(30, 45) for _ in range(7)], index)
weeks = [week1, week2, week3, week4]
print("Average temperature per week: ")
for week in weeks: print(round(week.mean(), 2))
print("\n")
print("AVG_TEMP: ")
avg_temp = (week1 + week2 + week3 + week4)/4
print(avg_temp,"\n")
print("Maximum temperature from the AVG_Temp Series: ")
print(avg_temp.max())
print("Minimum temperature from the AVG_Temp Series: ")
print(avg_temp.min())
print("Days where average temperature is above 35 degrees: ")
if avg_temp[avg_temp>35].count == 0: print("Temperature on none of the days was
above 35ºC")
else: print(avg_temp[avg_temp>35])
```

Output:

```
Average temperature per week:
37.71
```

```
36.29
40.86
34.0

AVG_TEMP:
Mon    38.00
Tue    39.25
Wed    35.50
Thu    38.75
Fri    36.00
Sat    33.25
Sun    39.75
dtype: float64

Maximum temperature from the AVG_Temp Series:
39.75
Minimum temperature from the AVG_Temp Series:
33.25
Days where average temperature is above 35 degrees:
Mon    38.00
Tue    39.25
Wed    35.50
Thu    38.75
Fri    36.00
Sun    39.75
dtype: float64
```

Question – 12

Q : Create a dataframe “temp” from the four series created above -week1, week2, week3 and week4.

The indexes should be ‘sun’, ‘mon’ ….’sat’ and columns should be week1, week2,week3 and week4.
Write a script to calculate:
1. Average temperatures for each day of the week i.e. average temperature for Monday, Tuesday etc.
2. Average temperature per week
3. Average temperature for whole month

Code:

```
import pandas as pd
from pandas import DataFrame as DF
temp = DF([week1, week2, week3, week4]).T
temp.rename(columns = {0:"week1",
    1:"week2",
    2:"week3",
    3:"week4"
}, inplace=True)

def avg_day():
    d = {}
    for x in temp.index:
        d[x] = temp.loc[x].mean()
    return pd.Series(d)

def avg_week():
    d={}
    for x in temp.columns:
        d[x] = round(temp[x].mean(),2)
    return pd.Series(d)

def avg_month():
    return round(avg_week().mean(),3)

def main():
    print("\nAverage temperatures for each day of the week")
    print(avg_day())
    print("\nAverage temperature per week")
    print(avg_week())
    print("\nAverage temperature per month")
    print(avg_month())
main()
```

Output:

```
Average temperatures for each day of the week
Mon    33.50
Tue    34.50
Wed    38.25
Thu    34.25
Fri    39.75
Sat    43.75
Sun    35.25
dtype: float64

Average temperature per week
week1    35.00
week2    39.86
week3    37.57
week4    35.71
dtype: float64

Average temperature per month
37.035
```

Question – 13

Q : Given a data frame 'data'

	Company	Quantity	Price
Printer	HCL	32	18000
Printer	Lenovo	7	12000
Scanner	HCL	12	14000
Monitor	Asus	46	15000
Scanner	Samsung	23	17000

Write a script to :

1. Find all rows with label “Printer”. Extract all columns.
2. List all the peripherals with quantity more than 30.
3. List true or false to signify if price is more than 15000 or not
4. List 2nd, 3rd and 4th rows.
5. List only the columns quantity and price using loc.
6. List only columns 1 and 2 (column indexes) using iloc.
7. List only rows with labels “Printer” and “Scanner” using loc.
8. Write statement to delete rows with labels “Printer” and “monitor”.
9. Write statement to delete columns with labels “company” and “price”

Code:

```
import pandas as pd
from pandas import DataFrame as DF
# DataFrame creation
d = {"Company": map(str, "HCL Lenovo HCL Asus Samsung".split()),
     "Quantity": map(int, "32 7 12 46 23".split()),
     "Price": map(int, "18000 12000 14000 15000 17000".split())}
data = DF(d, map(str, "Printer Printer Scanner Monitor Scanner".split()))
def main():
    print("Data: ")
    display(data)
    print("\n1. All rows with label 'Printer':")
    display(data.loc["Printer"])
    print("\n2. All the peripherals with quantity more than 30:")
    display(data[data["Quantity"] > 30])
    print("\n3. True or False to signify if price is more than 15000:")
    display(data["Price"]>15000)
    print("\n4. List the 2nd, 3rd and 4th rows:")
    display(data.iloc[[1,2,3,4]])
    print("\n5. List only the columns quantity and price using loc:")
    display(data.loc[:,["Quantity","Price"]])
    print("\n6. List only columns 1 and 2 (column indexes) using iloc: ")
    display(data.iloc[:,[0,1]])
    print("\n7. List only rows with labels “Printer” and “Scanner” using loc: ")
    display(data.loc[["Printer","Scanner"]])
    print("\n8. Delete rows with labels 'Printer' and 'monitor':")
    data.drop(["Printer", "Monitor"], inplace=True)
    display(data)
    print("\n9. Delete columns with labels 'company' and 'price':")
    data.drop(["Company", "Price"], 1, inplace=True)
    display(data)

main()
```

Output:

```
Data:

          Company  Quantity  Price

 Printer  HCL      32        18000

 Printer  Lenovo   7         12000

 Scanner  HCL      12        14000

 Monitor  Asus     50        15000

 Scanner  Samsung  23        17000


1. All rows with label 'Printer':

          Company  Quantity  Price

 Printer  HCL      32        18000

 Printer  Lenovo   7         12000


2. All the peripherals with quantity more than 30:
```

```
         Company  Quantity  Price

 Printer HCL      32        18000

 Monitor Asus     50        15000


3. True or False to signify if price is more than 15000:
Printer     True
Printer    False
Scanner    False
Monitor    False
Scanner     True
Name: Price, dtype: bool


4. List the 2nd, 3rd and 4th rows:

         Company  Quantity  Price

 Printer Lenovo   7         12000

 Scanner HCL      12        14000

 Monitor Asus     50        15000

 Scanner Samsung  23        17000


5. List only the columns quantity and price using loc:

         Quantity  Price

 Printer 32         18000

 Printer 7          12000

 Scanner 12         14000

 Monitor 50         15000

 Scanner 23         17000


6. List only columns 1 and 2 (column indexes) using iloc:
```

```
         Company  Quantity

Printer  HCL      32

Printer  Lenovo   7

Scanner  HCL      12

Monitor  Asus     50

Scanner  Samsung  23
```

7. List only rows with labels “Printer” and “Scanner” using loc:

```
         Company  Quantity  Price

Printer  HCL      32        18000

Printer  Lenovo   7         12000

Scanner  HCL      12        14000

Scanner  Samsung  23        17000
```

8. Delete rows with labels 'Printer' and 'monitor':

```
         Company  Quantity  Price

Scanner  HCL      12        14000

Scanner  Samsung  23        17000
```

9. Delete columns with labels 'company' and 'price':

```
         Quantity

Scanner  12

Scanner  23
```

Question – 14

Q : Write a python program to create a dataframe called “student” containing the following columns:
Rollno, name,Eng, Phy, Chem, Math, IP

Input the rollno, name and marks in 5 subjects.
Calculate the total and percentage.

All the newly created columns (totalmarks, per, grade) should be inserted in the dataframe. Print the final dataframe.

Calculate the grade as:

Percentage	Grade
91-100	A1
81-90	A2
71-80	B1
61-70	B2
51-60	C1
41-50	C2
33-40	D
< 33	E

Code:

```
import pandas as pd

try:
    roll, name, e, p, c, m, ip = input("Enter space separated value for ROLL NO., 
NAME, ENGLISH, PHYSICS, CHEMISTRY, MATHS, I.P. : ").split()
except Exception as err:
    print("[ERROR] ", str(err))
average =  round((int(e) + int(p) + int(c) + int(m) + int(ip))/5, 2)
total = average * 5

def get_grade(per):
    grades = list(map(str, "A1 A2 B1 B2 C1 C2 D E".split()[::-1]))
    marks = [(0,33),(33,40),(41,50),(51,60),(61,70),(71,80),(81,90),(91,100)]
    for i, tup in enumerate(marks):
        if tup[0] <= per <= tup[1]: return grades[i]
d={"Roll No.":roll,
   "Name":name,
   "English":e,
   "Physics":p,
   "Chemistry":c,
   "Mathematics":m,
   "I.P.":ip,
   "Total Marks (out of 500)" : total,
   "Total Percentage": average,
   "Grade" : get_grade(average)}
df = pd.DataFrame(d, index = [1])
index = 1
display(df)

cont=True
```

```
while cont:
    inp = input("Do you wish to add more entries? (yes/no) ")
    if inp.lower() == "no":
        cont=False
        print("Goodbye")
        break;
    elif inp.lower() != 'yes':
        print("Invalid Response. Please Try again.")
    try:
        roll, name, e, p, c, m, ip = input("Enter space separated value for ROLL 
NO., ENGLISH, PHYSICS, CHEMISTRY, MATHS, I.P. : ").split()
        average =  round((int(e) + int(p) + int(c) + int(m) + int(ip))/5, 2)
        total = average*5
        grade = get_grade(average)
    except Exception as err:
        print("[ERROR] ", str(err))
        print("Please try again.")
        pass
    index += 1
    df.at[str(index)] = [roll, name, e, p, c, m, ip, total, average, grade]
    display(df)
```

Output:

```
Enter space separated value for ROLL NO., NAME, ENGLISH, PHYSICS, CHEMISTRY, MATHS, I.P. : 123 A 90 
91 96 99 100

                                                                  Total
     Roll                                                                        Total
      No.  Name  English  Physics  Chemistry  Mathematics  I.P.  Marks (out   Percentage  Grade
                                                                   of 500)

 1    123     A       90       91         96           99   100       476.0         95.2     A1

Do you wish to add more entries? (yes/no) yes
Enter space separated value for ROLL NO., ENGLISH, PHYSICS, CHEMISTRY, MATHS, I.P. : 193 B 89 99 24 
79 91

                                                                  Total
     Roll                                                                        Total
      No.  Name  English  Physics  Chemistry  Mathematics  I.P.  Marks (out   Percentage  Grade
                                                                   of 500)

 1    123     A       90       91         96           99   100       476.0         95.2     A1

 2    193     B       89       99         24           79    91       382.0         76.4     B1

Do you wish to add more entries? (yes/no) yes
Enter space separated value for ROLL NO., ENGLISH, PHYSICS, CHEMISTRY, MATHS, I.P. : 1111 C 78 54 62 
27 43

                                                                  Total
     Roll                                                                        Total
      No.  Name  English  Physics  Chemistry  Mathematics  I.P.  Marks (out   Percentage  Grade
                                                                   of 500)

 1    123     A       90       91         96           99   100       476.0         95.2     A1

 2    193     B       89       99         24           79    91       382.0         76.4     B1

 3   1111     C       78       54         62           27    43       264.0         52.8     C1

Do you wish to add more entries? (yes/no) no
Goodbye
```

Question – 15

Q : Write a Python program to read a given CSV file having tab delimiter

Sample Output
country_id country_name region_id
AR Argentina 2
AU Australia 3
BE Belgium 1
BR Brazil 2
CA Canada 2

Code:

```
import pandas

df = pd.read_csv("sample.csv", sep="\t")

display(df)
```

Output:

```
   Country ID  Country Name  Region ID

0  AR          Argentina     2

1  AU          Australia     3

2  BE          Belgium       1

3  BR          Brazil        2

4  CA          Canada        2
```

Question – 16

Q : Write a Python program to read specific columns of a given CSV file and print the content of the columns.

Sample Output:
ID Department Name
10 Administration
20 Marketing
30 Purchasing
40 Human Resources
50 Shipping
60 IT

Code:

```
import pandas as pd

cols = ["Week 1","Week 3"]
df = pd.read_csv("sample2.csv", usecols=cols)
print(df)
```

Output:

```
   Week 1  Week 3
0      13      10
1      25      13
2      37      16
3      49      19
4      61      22
5      73      25
6      85      28
7      97      31
8     109      34
```

Question – 17

Q : Answer the following questions based on the table TRANSPORT.

j) Create the table and insert the following records

Rtno	Area_Covered	Capacity	No_of_ students	Distance	Transporter	Charges
1	VasantKunj	100	100	10	Shivam Tr.	10000.00
2	Rohini	80	80	10	Anand Tr	55000.75
3	Saket	50	50	30	Bhalla Co.	95000.50
4	Yamuna Vihar	120	120	35	Speed Tr	10000.00
5	Saket	100	100	20	Raj Tr.	80000.00
6	JanakPuri	40	40	10	Yadav Co.	60000.50

k) Display RtNo, Area_Covered and charges(without decimals) for all the school buses.

l) Display all the details from above table in ascending order of Area_Covered

m) Increase the charges of Yadav Co. by 1000.

n) Display Area_Covered in capitals and first three characters of the attribute Transporter.

o) Display the total charges according to each Area_covered.

p) Display the RtNo and Area Covered for the routes where No_of_students is more than the capacity.

q) Find outputs of the following queries:

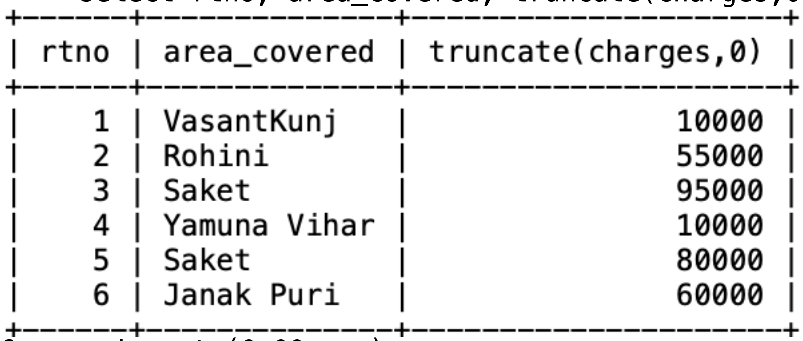
   2. Select Capacity, No_of_students from Transport where charges>60000;
   3. Select RtNo, charges/4 as “Quarterly Charges” from Transport;
   4. Select Transporter from Transport where Area_Covered like ‘S%’;

r) Remove column Distance.

```
(a)     mysql> create table Transport(
      -> Rtno int(2),
      -> Area_covered varchar(20),
      -> Capacity int(4),
      -> No_of_students int(4),
      -> Distance int(3),
      -> Transporter varchar(30),
      -> Charges float(8,2),
      -> primary key(Rtno));
   Query OK, 0 rows affected, 5 warnings (0.13 sec)

   mysql> insert into Transport values(1,"VasantKunj",100,100,10,"Shivam
   Tr.",10000.00);
   Query OK, 1 row affected (0.03 sec)

   mysql> insert into Transport values(2,"Rohini",80,80,10,"Anand Tr.",55000.75);
   Query OK, 1 row affected (0.01 sec)

   mysql> insert into Transport values(3,"Saket",50,50,30,"Bhalla Co.",95000.50);
```

```
Query OK, 1 row affected (0.00 sec)

mysql> insert into Transport values(4,"Yamuna Vihar",120,120,35,"Speed
Tr.",10000.00);
Query OK, 1 row affected (0.01 sec)

mysql> insert into Transport values(5,"Saket",100,100,20,"Raj Tr.",80000.00);
Query OK, 1 row affected (0.00 sec)

mysql> insert into Transport values(6,"Janak Puri",40,40,10,"Yadav Co.",60000.50);
Query OK, 1 row affected (0.00 sec)
```

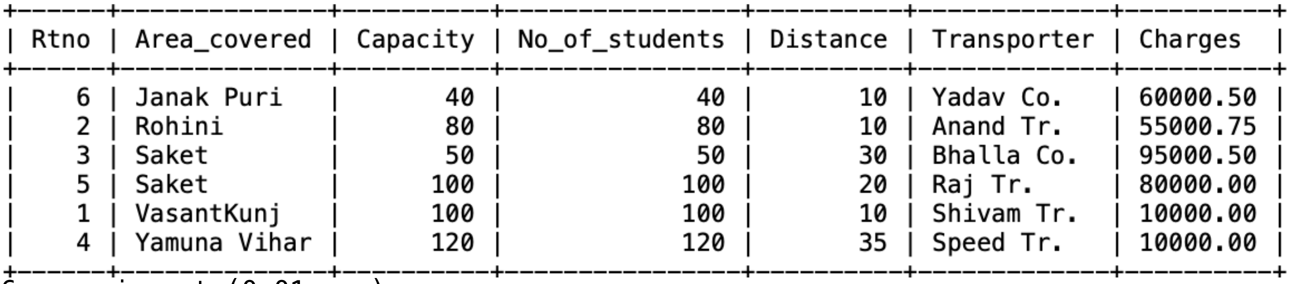
```
(b)     select rtno, area_covered, truncate(charges,0) from transport;
```



```
+------+--------------+---------------------+
| rtno | area_covered | truncate(charges,0) |
+------+--------------+---------------------+
|    1 | VasantKunj   |               10000 |
|    2 | Rohini       |               55000 |
|    3 | Saket        |               95000 |
|    4 | Yamuna Vihar |               10000 |
|    5 | Saket        |               80000 |
|    6 | Janak Puri   |               60000 |
+------+--------------+---------------------+
```

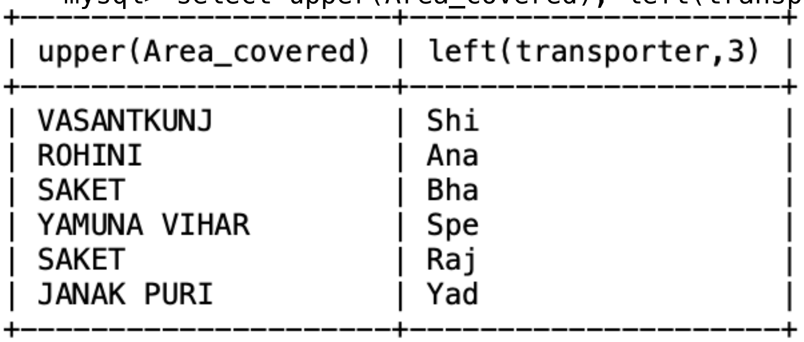
```
6 rows in set (0.00 sec)
```

```
(c)     mysql> select * from transport order by area_covered;
```



```
+------+--------------+----------+----------------+----------+-------------+----------+
| Rtno | Area_covered | Capacity | No_of_students | Distance | Transporter | Charges  |
+------+--------------+----------+----------------+----------+-------------+----------+
|    6 | Janak Puri   |       40 |             40 |       10 | Yadav Co.   | 60000.50 |
|    2 | Rohini       |       80 |             80 |       10 | Anand Tr.   | 55000.75 |
|    3 | Saket        |       50 |             50 |       30 | Bhalla Co.  | 95000.50 |
|    5 | Saket        |      100 |            100 |       20 | Raj Tr.     | 80000.00 |
|    1 | VasantKunj   |      100 |            100 |       10 | Shivam Tr.  | 10000.00 |
|    4 | Yamuna Vihar |      120 |            120 |       35 | Speed Tr.   | 10000.00 |
+------+--------------+----------+----------------+----------+-------------+----------+
```

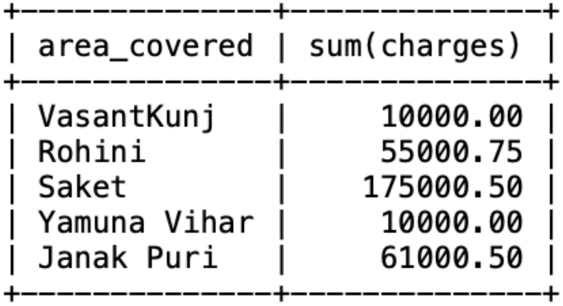
```
6 rows in set (0.01 sec)

(d)     mysql> update transport set charges = charges + 1000 where transporter = "Yadav
       Co.";
Query OK, 1 row affected (0.00 sec)
Rows matched: 1  Changed: 1  Warnings: 0

(e)    mysql> select upper(Area_covered), left(transporter,3) from transport;
```



```
+---------------------+---------------------+
| upper(Area_covered) | left(transporter,3) |
+---------------------+---------------------+
| VASANTKUNJ          | Shi                 |
| ROHINI              | Ana                 |
| SAKET               | Bha                 |
| YAMUNA VIHAR        | Spe                 |
| SAKET               | Raj                 |
| JANAK PURI          | Yad                 |
+---------------------+---------------------+
```

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

```
(f)    mysql> select area_covered, sum(charges) from transport group by area_covered;
```



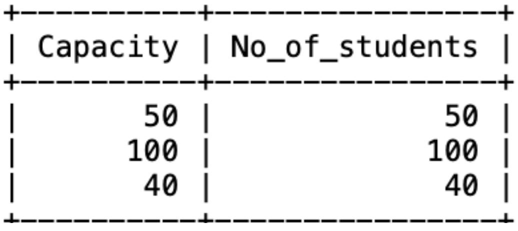
```
+--------------+--------------+
| area_covered | sum(charges) |
+--------------+--------------+
| VasantKunj   |     10000.00 |
| Rohini       |     55000.75 |
| Saket        |    175000.50 |
| Yamuna Vihar |     10000.00 |
| Janak Puri   |     61000.50 |
+--------------+--------------+
```

```
   5  rows in set (0.01 sec)
```

```
(g)    mysql> select rtno, area_covered from transport where no_of_students > capacity;
   Empty set (0.01 sec)
```

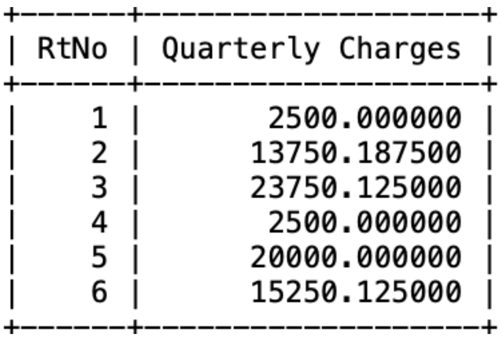
(h)

(I)



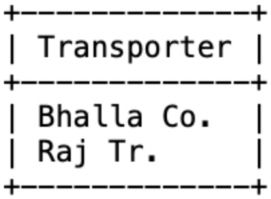
```
+----------+----------------+
| Capacity | No_of_students |
+----------+----------------+
|       50 |             50 |
|      100 |            100 |
|       40 |             40 |
+----------+----------------+
```

(II)



```
+------+-------------------+
| RtNo | Quarterly Charges |
+------+-------------------+
|    1 |       2500.000000 |
|    2 |      13750.187500 |
|    3 |      23750.125000 |
|    4 |       2500.000000 |
|    5 |      20000.000000 |
|    6 |      15250.125000 |
+------+-------------------+
```

(III)



```
+-------------+
| Transporter |
+-------------+
| Bhalla Co.  |
| Raj Tr.     |
+-------------+
```

```
(i)    mysql> alter table transport drop Distance;
       Query OK, 0 rows affected (0.38 sec)
       Records: 0  Duplicates: 0  Warnings: 0
```

Question – 18

Q : Make the following two tables ITEM and CUSTOMER and write the commands for the following

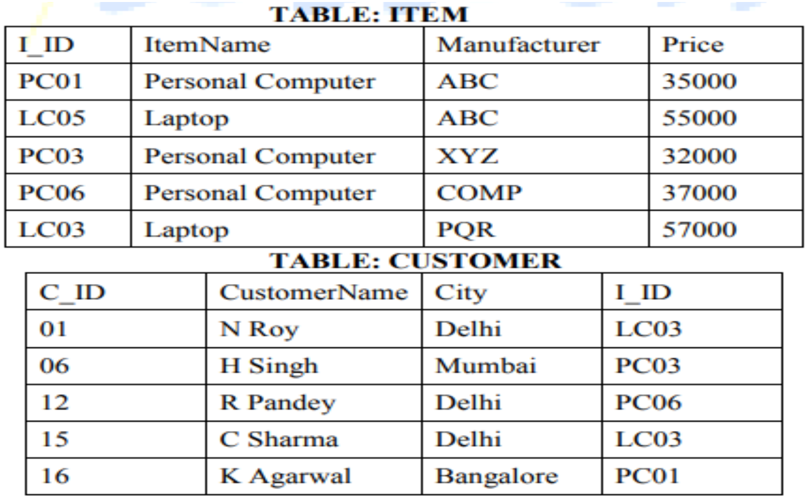


TABLE: ITEM

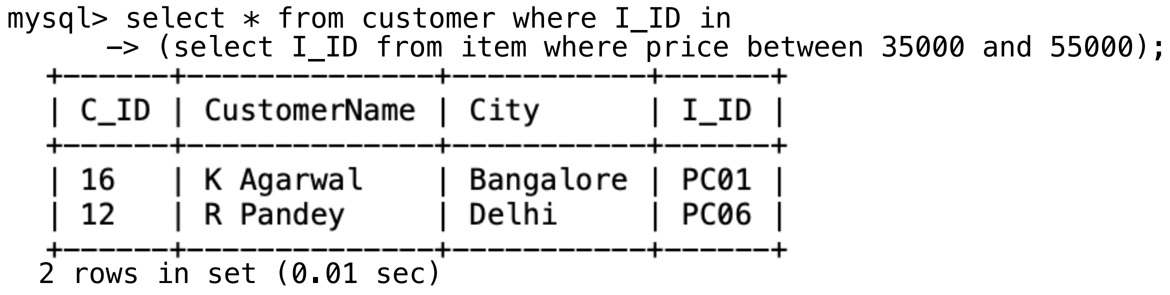
I_ID	ItemName	Manufacturer	Price
PC01	Personal Computer	ABC	35000
LC05	Laptop	ABC	55000
PC03	Personal Computer	XYZ	32000
PC06	Personal Computer	COMP	37000
LC03	Laptop	PQR	57000

TABLE: CUSTOMER

C_ID	CustomerName	City	I_ID
01	N Roy	Delhi	LC03
06	H Singh	Mumbai	PC03
12	R Pandey	Delhi	PC06
15	C Sharma	Delhi	LC03
16	K Agarwal	Bangalore	PC01

a) To display the details of customers of those items whose Price is in the range of 35000 to 55000 (Both values included).
b) To display the average price of items based on cities of their customers
c) To display the CustomerName, City from table Customer and ItemName from table Item.
d) To display ItemName and Price from table Item for the customers living in Mumbai
e) To increase the Price of all Items by 1000 in the table Item for all personal computers
f) To Display details of customers who purchased the costliest item
g) Add one more column STOCK as varchar (20) to ITEM table

(a)



```
mysql> select * from customer where I_ID in
    -> (select I_ID from item where price between 35000 and 55000);
+------+--------------+-----------+------+
| C_ID | CustomerName | City      | I_ID |
+------+--------------+-----------+------+
| 16   | K Agarwal    | Bangalore | PC01 |
| 12   | R Pandey     | Delhi     | PC06 |
+------+--------------+-----------+------+
2 rows in set (0.01 sec)
```

(b)

```
mysql> select city, avg(price) from item,customer where item.I_ID = customer.I_ID
group by City;
```

```
   +-----------+------------+
   | city      | avg(price) |
   +-----------+------------+
   | Delhi     | 50333.3333 |
   | Bangalore | 35000.0000 |
   | Mumbai    | 32000.0000 |
   +-----------+------------+
   3 rows in set (0.01 sec)
```

```
(c)   mysql> select customerName, city, itemName from item, customer where customer.I_ID
      = item.I_id;
   +--------------+-----------+------------------+
   | customerName | city      | itemName         |
   +--------------+-----------+------------------+
   | N Roy        | Delhi     | Laptop           |
   | C Sharma     | Delhi     | Laptop           |
   | K Agarwal    | Bangalore | Personal Computer |
   | H Singh      | Mumbai    | Personal Computer |
   | R Pandey     | Delhi     | Personal Computer |
   +--------------+-----------+------------------+
   5 rows in set (0.01 sec)
```

```
(d)   mysql> select itemName, price, city from item, customer where customer.I_ID =
      item.I_id and city
         = "Mumbai";
   +-------------------+-------+--------+
   | itemName          | price | city   |
   +-------------------+-------+--------+
   | Personal Computer | 32000 | Mumbai |
   +-------------------+-------+--------+
   1 row in set (0.00 sec)
```

```
(e)   mysql> update item set price = price + 1000 where ItemName = "Personal Computer";
         Query OK, 3 rows affected (0.01 sec)
         Rows matched: 3  Changed: 3  Warnings: 0
```

```
(f)   mysql> select customer.* from customer, item where item.I_ID = customer.I_ID and
      item.price = (select max(price) from item);
   +------+--------------+-------+------+
   | C_ID | CustomerName | City  | I_ID |
   +------+--------------+-------+------+
   | 01   | N Roy        | Delhi | LC03 |
   | 15   | C Sharma     | Delhi | LC03 |
   +------+--------------+-------+------+
   2 rows in set (0.09 sec)
```

```
(g)   mysql> alter table item add Stock varchar (20);
         Query OK, 0 rows affected (0.14 sec)
         Records: 0  Duplicates: 0  Warnings: 0
```