**COMPUTER SCIENCE WITH PYTHON**

**PRACTICAL FILE**

NAME:

CLASS: XII

SECTION:

PRACTICAL FILE- COMPUTER SCIENCE (083)

LIST OF PRACTICALS (2025-26) CLASS-XII

Programming Language: Python

|  |  |
| --- | --- |
| **S.**  **No.** | **NAME OF PRACTICAL** |
| 1 | Write a Program using functions to enter the string and to check if it’s palindrome |
| 2 | Write a function that takes a character and returns True if it is a vowel , False otherwise. |
| 3 | Write a program that inputs two tuples and creates a third that contains all elements of the first followed by all the elements of the second. |
| 4 | Write a program using function to create a dictionary, which accepts the marks in five different subjects with subject as key and marks as value. Display the dictionary and the subjects getting marks 90 and above. |
| 5 | Write a program using function that takes one argument (a positive integer ) and reports if the argument is prime or not. |
| 6 | Write a program to generate random numbers between 1 to 6 and check whether a user won a  lottery or not. |
| 7 | Write a program using function to generate Fibonacci series up to n. Pass the value of ‘n’ to a function and display all the numbers of the series in the function as a list elements. |
| 8 | Write a program to copy the records of that students having marks greater than 95 from binary file "marks dat" into the "toppers.dat" file. |
| 9 | Write a program that reads a text file and then creates a new file where each character’s case is inverted. |
| 10 | Read a text file and display the number of vowels/ consonants/ uppercase/ lowercase characters in the file. |
| 11 | Write a function in PYTHON to count and display the number of words starting with alphabet ‘A’ or ‘a’ present in a text file “LINES.TXT”.  Example:  If the file “LINES.TXT” contains the following lines,  “A boy is playing there. There is a playground. An aeroplane is in the sky.  Are you getting it?”  The function should display the output as 5 . |
| 12 | Consider a binary file “Emp.dat” containing details such as empno: ename:salary(separator‘:’).Write a python function to display details of those employees who are earning between 20000 and 40000. |
| 13 | Write a program that will write a string in binary file "school.dat" and display the words of the string in reverse order. |
| 14 | Write a program to read data from a text file DATA.TXT, and display word, which have maximum/minimum characters. |
| 15 | Write the function definition for WORD4CHAR() in PYTHON to read the content of a text file FUN.TXT, and display all those words, which have four characters in it.  Example:  If the content of the file Fun.TXT is as follows:  “When I was a small child, I used to play in the garden with my grand mom. Those days were amazingly funful and I remember all the moments of that time”  The function WORD4CHAR() should display the following:  “When used play with days were that time” |
| 16 | Write a definition for function BUMPER() in PYTHON to read each object of a binary file GIFTS.DAT, find and display details of those gifts, which have remarks as “ON DISCOUNT”. Assume that the file GIFTS.DAT is created with the help of lists of following type:  (ID, Gift, Remarks, Price**)** |
| 17 | Create a binary file “batsman.dat” with the fields:b\_No,Name,runs,rank and write a function count() in python which accepts the rank from user and counts the number of the file having that rank. |
| 18 | Create a binary file with roll number, name and marks. Input a roll number and update the marks. |
| 19 | Following is the structure of each record in a data file named ”PRODUCT.DAT”.  {"prod\_code":value, "prod\_desc":value, "stock":value}  The values for prod\_code and prod\_desc are strings, and the value for stock is an integer. Write a function in PYTHON to update the file with a new value of stock. The stock and the product\_code, whose stock is to be updated, are to be input during the execution of the function. |
| 20 | Write a Python program to get 10 items’ details (itemno,name,price,category) from the user and create a CSV file(Items.csv). |
| 21 | Write a menu-based program to perform the operation on stack in python. |
| 22 | Write a program to connect Python with MySQL using database connectivity and perform the  following operations on data in database: Fetch, Update and delete the data |
| 23 | **Consider the following tables STORE and SUPPLIERS and answer (a) and (b) parts of this question:**  **Table: - Store**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **ItemNo.** | **Iteam** | **Scode** | **Qty** | **Rate** | **LastBuy** | | 2005 | Sharpener Classic | 23 | 60 | 8 | 31-Jun-09 | | 2003 | Ball Pen 0.25 | 22 | 50 | 25 | 01-Feb-10 | | 2002 | Gel Pen Premium | 21 | 150 | 12 | 24-Feb-10 | | 2006 | Gel Pen Classic | 21 | 250 | 20 | 11-Mar-09 | | 2001 | Eraser Small | 22 | 220 | 6 | 19-Jan-09 | | 2004 | Eraser Big | 22 | 110 | 8 | 02-Dec-09 | | 2009 | Ball Pen 0.5 | 21 | 180 | 18 | 03-Nov-09 |   **Table:- Supplier**   |  |  | | --- | --- | | **Scode** | **Sname** | | 21 | Premium Stationers | | 23 | Soft Plastics | | 22 | Tetra Supply |   **A.Write SQL commands for the following statements:**   1. To display details of all the items in the Store table in ascending order of LastBuy. 2. To display ItemNo and Item name of those items from Store table, whose Rate is more than 15 Rupees. 3. To display the details of those items whose Supplier code (Scode) is 22 or Quantity in Store (Qty) is more than 110 from the table Store. 4. **To display Minimum Rate of items for each Supplier individually as per Scode from the table Store**   **B.Give the output of the following SQL queries:**   1. SELECT COUNT(DISTINCT Scode) FROM Store; 2. SELECT Rate\*Qty FROM Store WHERE ItemNo=2004; 3. SELECT Item,Sname FROM Store S, Suppliers P WHERE S.Scode=P.Scode AND ItemNo=2006;   (iv)SELECT MAX(LastBuy) FROM Store; |
| 24 | **Consider the following table Item and Customer.**  **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 | Banglore | PC01 | |  |  |  |  | |
|  | 1. **Write SQL commands for the following statements:** 2. To display the details of those Customer whose City is Delhi. 3. To display the details of Item whose Price is in the range of 3500 to 55000 (Both values included). 4. To display the CustomerName, City from table Customer, and ItemName and Price from table Item with their corresponding matching I\_ID. 5. To increase the Price of all Items by 1000 in the table Item. 6. **Give the output of the following SQL queries:** 7. SELECT DISTINCT City FROM Customer; 8. SELECT ItemName, Max(Price), Count(\*) FROM Item GROUP BY ItemName; 9. SELECT CustomerName, Manufacturer From Item, Customer WHERE Item.I\_Id=Customer.I\_Id; 10. SELECT ItemName, Price \* 100 FROM Item WHERE Manufacturer = ‘ABC’; |
| 25 | **Consider the following tables.**  **TABLE : SENDER**   |  |  |  |  | | --- | --- | --- | --- | | **SenderID** | **SenderName** | **SenderName** | **SenderCity** | | ND01 | R Jain | 2,ABC Appts | New Delhi | | MU02 | H Sinha | 12, Newtown | Mumbai | | MU15 | S Jha | 27/A, Park Street | Mumbai | | ND50 | T Prasad | 122-K, SDA | New Delhi |   **TABLE : RECIPIENT**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **RecID** | **SenderID** | **RecName** | **RecAddress** | **ReCity** | | KO05 | ND01 | R Bajpayee | 5, Central Avenue | Kolkata | | ND08 | MU02 | S Mohan | 116, A vihar | New Delhi | | MU19 | ND01 | H singh | 2a, Andheri east | Mumbai | | MU32 | MU15 | P K Swamy | B5, c S Terminus | Mumbai | | ND48 | ND50 | S Tirupathi | 13, B1 d, Mayur vihar | New Delhi |  1. **Write SQL commands for the following statements:** 2. To display the names of all Senders from Mumbai. 3. To display the RecID, SenderName, SenderAddress, RecName, RecAddess for every Recipient 4. To display Recipient detail in ascending order of RecName 5. To display number of Recipients from each city 6. **Give the output of the following SQL queries:** 7. SELECT DISTINCT Sendercity FROM Sender; 8. SELECT A.SenderName, B.RecName FROM Sender A, Recipient B WHERE A.SenderID=B.SenderID AND B.RecCity=’Mumbai’; 9. SELECT RecName, RecAddress FROM Recipient WHERE recCity NOT IN(‘Mumbai’, ‘Kolkata’); 10. SELECT RecID, RecName FROM Recipient WHERE SenderID=’MU02’ OR SenderID=’ND50’; |
| 26 | **Consider the following tables FURNITURE and ARRIVALS: write SQL commands for the questions (a) to (f) and give outputs for SQL queries (i) to (iv) :**  **TABLE : FURNITURE**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **NO** | **ITEMNAME** | **TYPE** | **DATEOFSTOCK** | **PRICE** | **DISCOUNT** | | 1 | White lotus | Double Bed | 23/02/02 | 30000 | 25 | | 2 | Pink feather | Baby cot | 20/01/02 | 7000 | 20 | | 3 | Dolphin | Baby cot | 19/02/02 | 9500 | 20 | | 4 | Decent | Office Table | 01/01/02 | 25000 | 30 | | 5 | Comfort zone | Double Bed | 12/01/02 | 25000 | 25 | | 6 | Donald | Baby cot | 24/02/02 | 6500 | 15 | | 7 | Royal Finish | Office Table | 20/02/02 | 18000 | 30 | | 8 | Royal tiger | Sofa | 22/02/02 | 31000 | 30 | | 9 | Econo sitting | Sofa | 13/12/01 | 9500 | 25 | | 10 | Eating Paradise | Dining Table | 19/02/02 | 11500 | 25 |   **TABLE : ARRIVALS**   |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | **NO** | **ITEMNAME** | **TYPE** | **DATEOFSTOCK** | **PRICE** | **DISCOUNT** | | 11 | Wood Comfort | Double Bed | 23/03/03 | 25000 | 25 | | 12 | Old Fox | Sofa | 20/02/03 | 17000 | 20 | | 13 | Micky | Baby cot | 21/02/03 | 7500 | 15 |  1. **Write SQL commands for the following statements:** 2. To show all information about the Baby cots from the FURNITURE table. 3. To list the ITEMNAME which are priced at more than 15000 from the FURNITURE table. 4. To list ITEMNAME and TYPE of those items, in which DATEOFSTOCK is before 22/01/02 from the FURNITURE table in descending order of ITEMNAME. 5. To display ITEMNAME and DATEOFSTOCK of those items, in which the DISCOUNT percentage is more than 25 form FURNITURE table. 6. To count the number of items, whose TYPE is “Sofa” from FURNITURE table. 7. To insert a new row in the ARRIVALS table with the following data:    1. 14, ‘Velvet touch’, ‘Double bed’, {25/03/03}, 25000,30 8. **Give the output of following SQL statement:**   *NOTE : Outputs of the below mentioned queries should be based on original data given in both the tables, i.e., without considering the insertion done in (f) part of this question :*   1. Select COUNT (distinct TYPE) from FURNITURE; 2. Select MAX(DISCOUT) form FURNITURE, ARRIVALS; 3. Select AVG(DISCOUT) form FURNITURE where TYPE = ‘Baby cot’; 4. Select SUM(PRICE) from FURNITURE where DATEOFSTOCK<{12/02/02}; |
| 27 | **Study the following tables DOCTOR and SALARY and write SQL commands for the questions (i) to (iv) and give outputs for SQL queries (v) to (vi) :**  **TABLE : DOCTOR**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ID** | **NAME** | **DEPT** | **SEX** | **EXPERIENCE** | | 101 | John | ENT | M | 12 | | 104 | Smith | ORTHOPEDIC | M | 5 | | 107 | George | CARDIOLOGY | M | 10 | | 114 | Lara | SKIN | F | 3 | | 109 | K George | MEDICINE | F | 9 | | 105 | Johnson | ORTHOPEDIC | M | 10 | | 117 | Lucy | ENT | F | 3 | | 111 | Bill | MEDICINE | F | 12 | | 130 | Morphy | ORTHOPEDIC | M | 15 | |
|  | **TABLE : SALARY**   |  |  |  |  | | --- | --- | --- | --- | | **ID** | **BASIC** | **ALLOWANCE** | **CONSULTATION** | | 101 | 12000 | 1000 | 300 | | 104 | 23000 | 2300 | 500 | | 107 | 32000 | 4000 | 500 | | 114 | 12000 | 5200 | 100 | | 109 | 42000 | 1700 | 200 | | 105 | 18900 | 1690 | 300 | | 130 | 21700 | 2600 | 300 | |
|  | 1. **Write SQL commands for the following statements:**    * 1. Display NAME of all doctors who are in “MEDICINE” having more than 10 year experience from the table DOCTOR.      2. Display the average salary of all doctors working in “ENT” department using the tables DOCTOR and SALARY. Salary=BASIC + ALLOWANCE.      3. Display the minimum ALLOWANCE of female doctors.      4. Display the highest consultation fee among all male doctor |
|  | 1. **Give the output of following SQL statement:**    * 1. SELECT count(\*) from DOCTOR where SEX=”F”.      2. SELECT NAME, DEPT, BASIC from DOCTOR Salary WHERE DEPT=”ENT” AND DOCTORID=SALARY.ID |

Q1 #Write a Program using functions to enter the string and to check if it’s palindrome

Code

def check\_palindrome(s):

return s == s[::-1]

s = input("Enter a string: ")

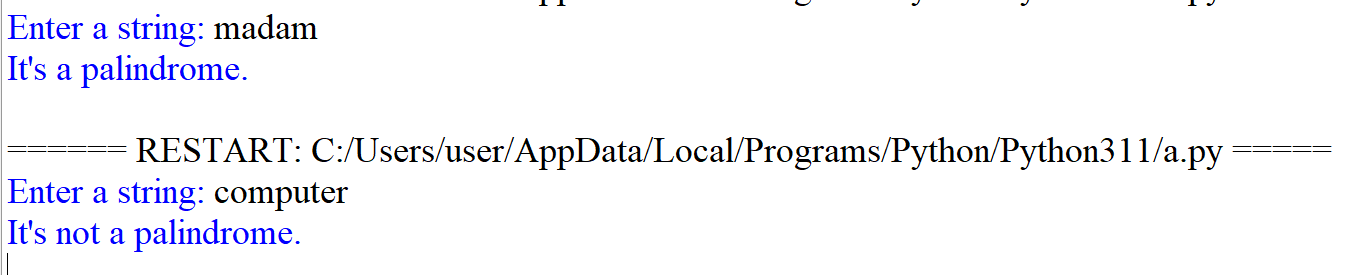
if check\_palindrome(s):

print("It's a palindrome.")

else:

print("It's not a palindrome.")

OUTPUT



Q2 #Write a function that takes a character and returns True if it is a vowel , False otherwise.

Code

def checkvowel(x):

if (x == 'a' or x == 'e' or x == 'i' or x == 'o' or x == 'u'):

print("Vowel")

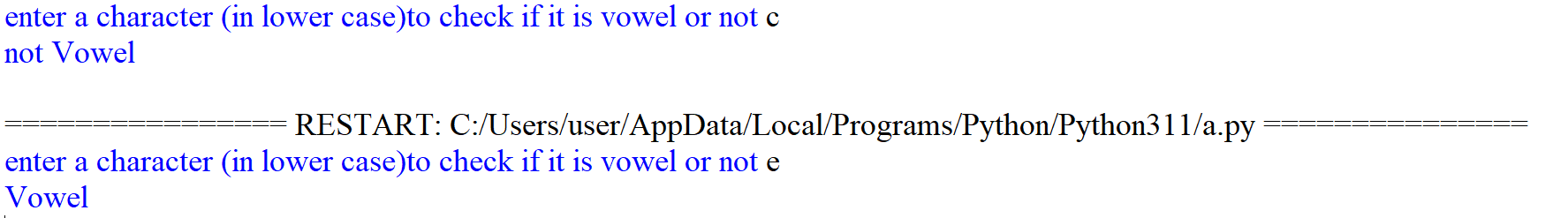
else:

print("not Vowel")

ch=input('enter a character (in lower case)to check if it is vowel or not ')

checkvowel(ch)

OUTPUT



Q3 #Write a program that inputs two tuples and creates a third that contains all elements of the first followed by all the elements of the second.

Code

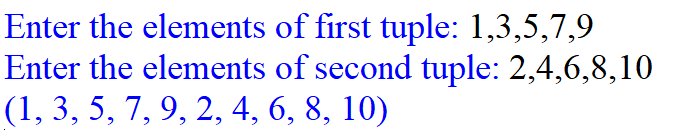
tup1 = eval(input("Enter the elements of first tuple: "))

tup2 = eval(input("Enter the elements of second tuple: "))

tup3 = tup1 + tup2

print(tup3)

OUTPUT



Q4 #Write a program using function to create a dictionary, which accepts the marks in five different subjects with subject as key and marks as value. Display the dictionary and the subjects getting marks 90 and above.

Code

def display\_marks90(marks\_dict):

for i in marks\_dict:

if marks\_dict[i]>90:

print(i,marks\_dict[i])

marks\_dict = {}

for i in range(5):

subject = input("Enter the name of subject : ")

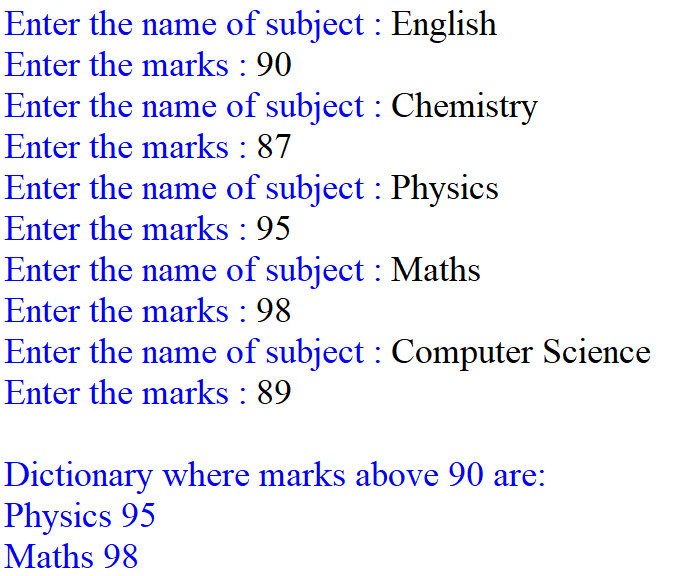
marks = int(input("Enter the marks : "))

marks\_dict[subject] = marks

print("\nDictionary where marks above 90 are:")

display\_marks90(marks\_dict)

OUTPUT



Q5 #Write a program using function that takes one argument (a positive integer ) and reports if the argument is prime or not.

Code

def check\_prime(n):

if (n == 1):

return False

elif (n == 2):

return True

else:

for x in range(2, n):

if (n % x == 0):

return False

return True

n=int(input('enter any no'))

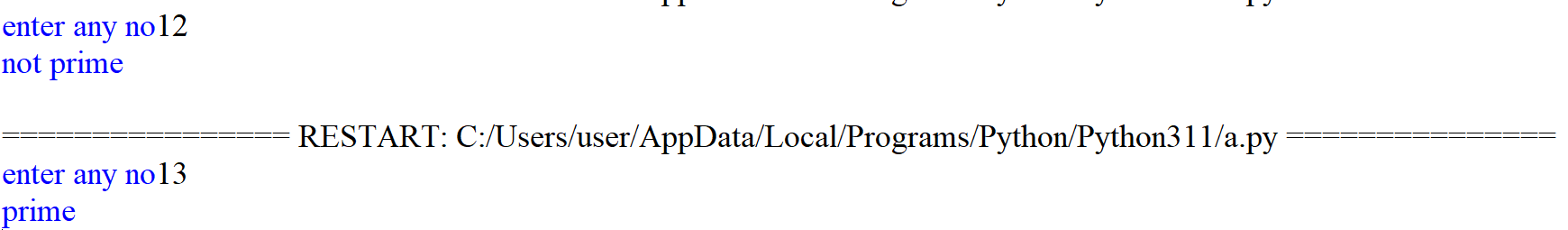
if check\_prime(n):

print('prime')

else:

print('not prime')

OUTPUT



Q6 #Write a program to generate random numbers between 1 to 6 and check whether a user won a lottery or not.

Code

import random

def generate\_lottery\_number(guess):

# Generate a random 6-digit lottery number

lottery\_number = random.randint(1, 6)

print('lottery number',lottery\_number)

if guess>=1 and guess<=6:

if guess==lottery\_number:

print('won lottery')

else:

print('not won lottery')

else:

print('Invalid, enter no between 1 and 6')

guess\_no=int(input('enter any no between 1 and 6'))

generate\_lottery\_number(guess\_no)

OUTPUT



Q7 #Write a program using function to generate Fibonacci series up to n. Pass the value of ‘n’ to a

#function and displayall the numbers of the series in the function as a list elements.

Code

def fibo(n):

n1, n2 = 0, 1

print("Fibonacci sequence:")

print(n1)

print(n2)

count = 0

while count < n-2:

c=n1+n2

print(c)

n1=n2

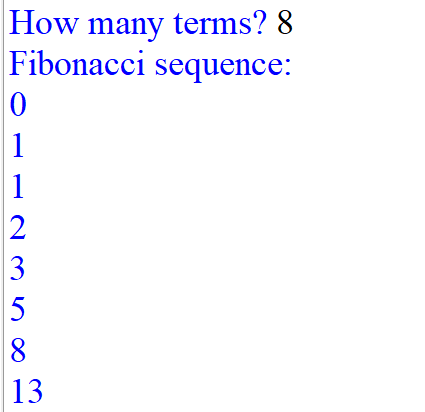
n2=c

count+=1

nterms = int(input("How many terms? "))

fibo(nterms)

OUTPUT



Q8 # Write a program to copy the records of that students having marks greater than 95 from binary file "marks dat" into the "toppers.dat" file.

Code

import pickle

def fun():

f=open("marks.DAT","rb")

f1=open("toppers.DAT","wb")

while True:

try:

r=pickle.load(f)

if(r[2]>95:

pickle.dump(r,f1)

except:

break

print(c)

f.close()

f1.close()

Q9 #Write a program that reads a text file and then creates a new file where each character’s case is inverted.

Code

f=open('file.txt','w')

f.write('hello world')

f.close()

f1 = open("output1.txt", "w")

with open("file.txt", "r") as myfile:

data = myfile.read()

for i in data.split():

data\_1 = i[::-1]

f1.write(data\_1)

f1.write(' ')

f1.close()

with open('output1.txt','r') as f2:

print(f2.read())

OUTPUT



Q10 #Read a text file and display the number of vowels/ consonants/ uppercase/ lowercase characters in the file.

Code

f=open('file.txt','w')

f.write('Hello World')

f.close()

v=0

upp=0

low=0

c=0

with open("file.txt", "r") as f1:

d=f1.read()

print('file content is: ', d)

for i in d:

if i.islower():

low=low+1

elif i.isupper():

upp=upp+1

if i.isalpha():

if i.lower() in 'aeiou':

v=v+1

else:

c=c+1

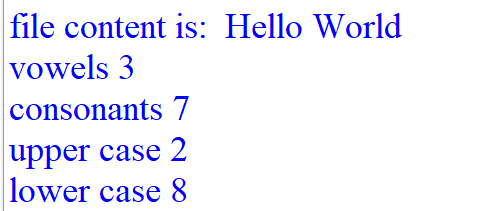
print('vowels',v)

print('consonants',c)

print('upper case',upp)

print('lower case',low)

OUTPUT



Q11 #Write a function in PYTHON to count and display the number of words starting with alphabet ‘A’ or ‘a’ present in a text file “LINES.TXT”.

Code

f=open('LINES.TXT','w')

f.write('A boy is playing there. There is a playground. An aeroplane is in the sky.\nAre you getting it?')

f.close()

with open("LINES.TXT", "r") as f1:

d=f1.read()

print('file content is: ', d)

count=0

for i in d.split():

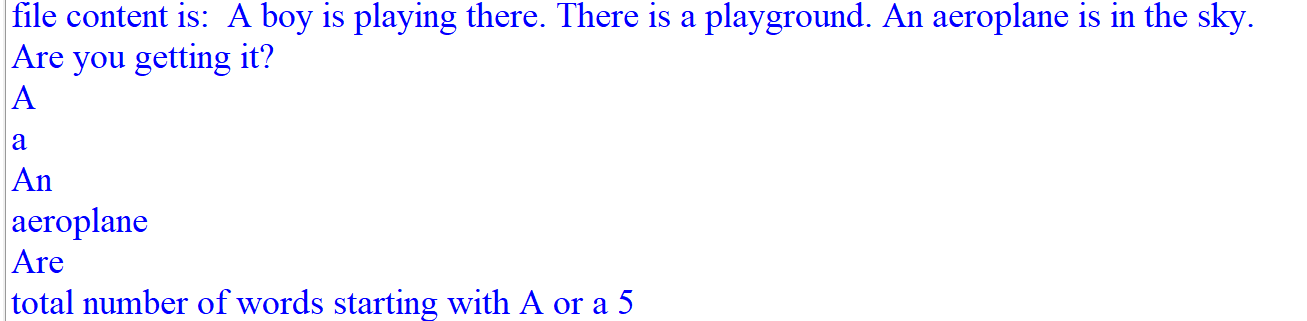
if i[0]=='A' or i[0]=='a':

print(i)

count=count+1

print('total number of words starting with A or a',count)

OUTPUT



Q12 #Consider a binary file “Emp.dat” containing details such as empno: ename:salary(separator‘:’).Write a python function to display details of those employees who are earning between 20000 and 40000.

Code

import pickle

def add():

f=open('EMP.dat','wb')

n=int(input('how many records to add'))

for i in range(n):

en=int(input('enter employee no'))

ename=input('enter employee name')

sal=int(input('enter employee salary'))

pickle.dump([en,ename,sal],f)

f.close()

def show():

f=open('EMP.dat','rb')

while True:

try:

d=pickle.load(f)

if d[2]>=20000 and d[2]<=40000:

print(d)

except:

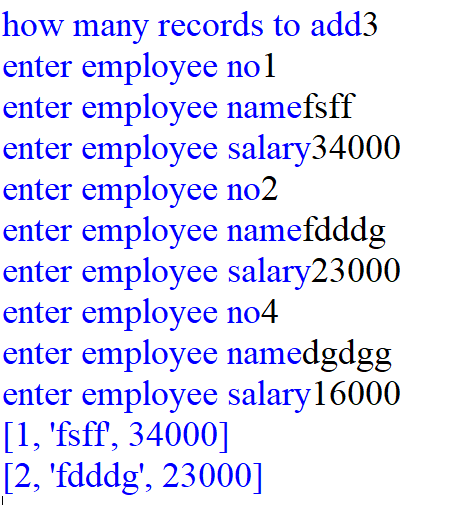
pass

f.close()

add()

show()

OUTPUT



Q13 #Write a program that will write a string in binary file "school.dat" and display the words of the string in reverse order.

Code

import pickle

def add():

f=open('school.dat','wb')

n=int(input('how many strings to add'))

for i in range(n):

s=input('enter string')

pickle.dump(s,f)

f.close()

def show():

f=open('school.dat','rb')

while True:

try:

d=pickle.load(f)

for i in d.split():

print(i[::-1],end=' ')

except:

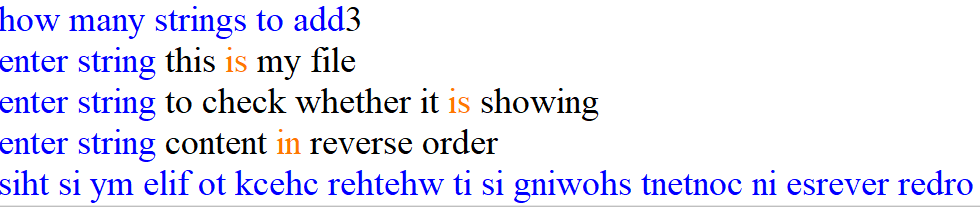
pass

f.close()

add()

show()

OUTPUT



Q14 #Write a program to read data from a text file DATA.TXT, and display word, which have maximum/minimum characters.

Code

f=open('DATA.TXT','w')

f.write('A boy is playing there There is a playground An aeroplane is in the sky Are you getting it')

f.close()

with open("DATA.TXT", "r") as f1:

g=[]

d=f1.read()

print('file content is',d)

for i in d.split():

g.append(len(i))

m=max(g)

n=min(g)

y=[]

z=[]

for i in d.split():

if len(i)==m:

y.append(i)

elif len(i)==n:

z.append(i)

else:

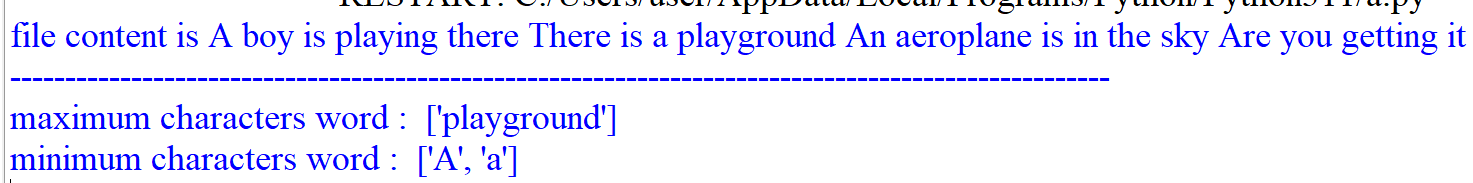
continue

print('-'\*100)

print('maximum characters word : ',y)

print('minimum characters word : ',z)

OUTPUT



Q15 #Write the function definition for WORD4CHAR() in PYTHON to read the content of a text file FUN.TXT, and display all those words, which have four characters in it.

Example:

If the content of the file Fun.TXT is as follows:

“When I was a small child, I used to play in the garden with my grand mom. Those days were amazingly funful and I remember all the moments of that time”

The function WORD4CHAR() should display the following:

“When used play with days were that time”

Code:

f=open('FUN.TXT','w')

f.write('When I was a small child, I used to play in the garden with my grand mom. Those days were \namazingly funful and I remember all the moments of that time?')

f.close()

def WORD4CHAR():

with open("FUN.TXT", "r") as f1:

d=f1.read()

print('file content is',d)

print('-'\*100)

print('All these words, have four characters')

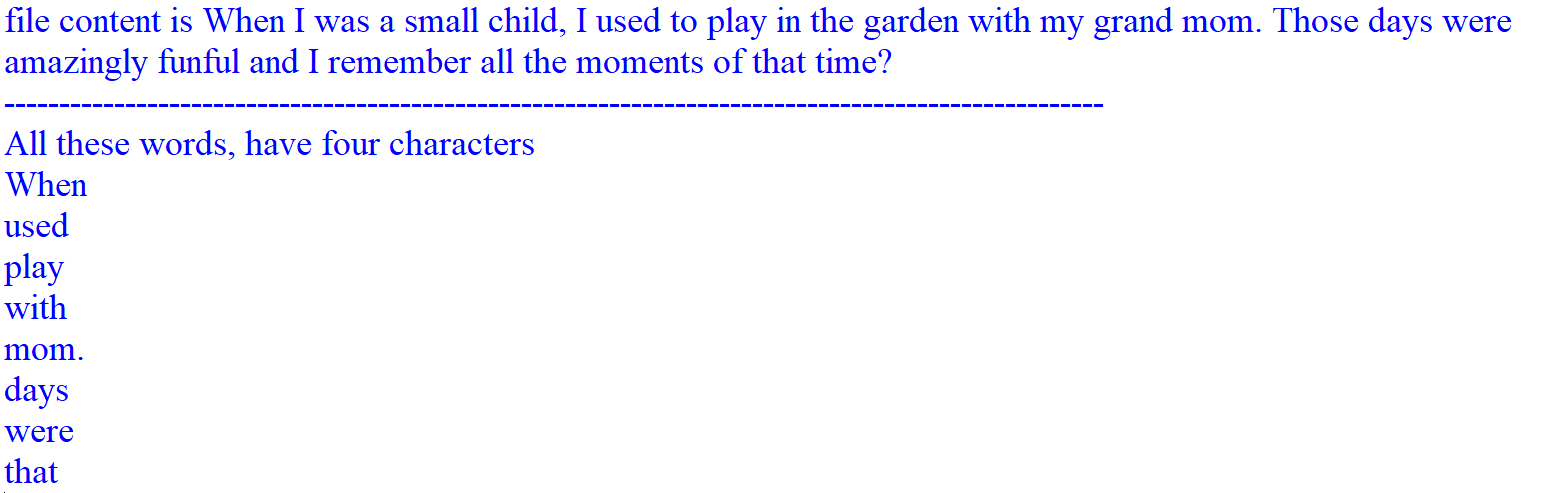
for i in d.split():

if len(i)==4:

print(i)

WORD4CHAR()

OUTPUT



Q16 # Write a definition for function BUMPER() in PYTHON to read each object of a binary file GIFTS.DAT, find and display details of those gifts, which have remarks as “ON DISCOUNT”. Assume that the file GIFTS.DAT is created with the help of lists of following type:

(ID, Gift, Remarks, Price**)**

Code

import pickle

def add():

with open("GIFTS.DAT", "wb") as f:

while True:

Gid=int(input('enter gift id'))

gname=input('enter gift name')

rem=input('enter remarks')

pr=int(input('enter gift price'))

pickle.dump([Gid,gname,rem,pr],f)

ch=input('want to add more')

if ch=='n' or ch=='N':

break

def BUMPER():

with open("GIFTS.DAT", "rb") as f:

try:

while True:

gift = pickle.load(f)

if gift[2] == "ON DISCOUNT":

print(gift)

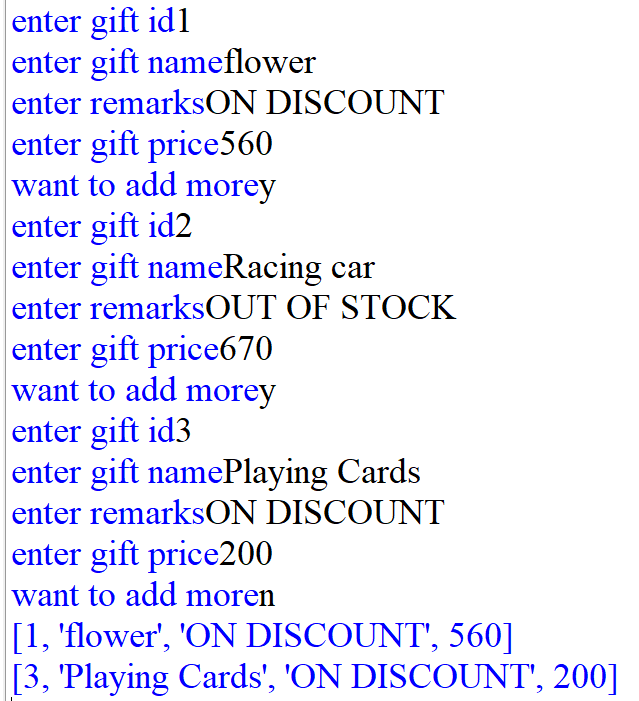
except EOFError:

pass

add()

BUMPER()

OUTPUT



Q17 # Create a binary file “batsman.dat” with the fields:b\_No,Name,runs,rank and write a function count() in python which accepts the rank from user and counts the number of the file having that rank.

Code

import pickle

def add():

with open("batsman.dat", "wb") as f:

while True:

b\_no=int(input('enter batsman id'))

name=input('enter name')

run=input('enter runs scored by him')

rank=int(input('enter his rank'))

pickle.dump([b\_no,name,run,rank],f)

ch=input('want to add more')

if ch=='n' or ch=='N':

break

def count\_rank(rank\_val):

with open("batsman.dat", "rb") as f:

count = 0

try:

while True:

b = pickle.load(f)

if b[3] == rank\_val:

count += 1

except EOFError:

pass

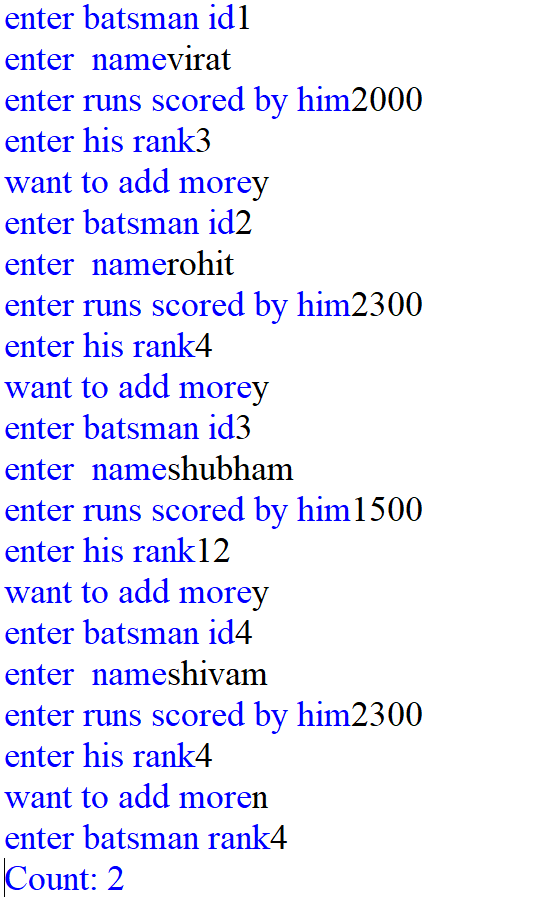
print("Count:", count)

add()

rank=int(input('enter batsman rank'))

count\_rank(rank)

OUTPUT



Q18 # Create a binary file with roll number, name and marks. Input a roll number and update the marks.

Code

import pickle

def add():

with open("student.dat", "wb") as f:

while True:

rno=int(input('enter rollno'))

name=input('enter name')

marks=int(input('enter marks'))

pickle.dump([rno,name,marks],f)

ch=input('want to add more')

if ch=='n' or ch=='N':

break

def up(roll):

new\_data = []

with open("student.dat", "rb") as f:

try:

while True:

rec = pickle.load(f)

if rec[0] == roll:

rec[2] = int(input("Enter new marks: "))

new\_data.append(rec)

else:

new\_data.append(rec)

except EOFError:

pass

with open("student.dat", "wb") as f:

for rec in new\_data:

pickle.dump(rec, f)

def show():

with open("student.dat", "rb") as f:

try:

while True:

rec = pickle.load(f)

print(rec)

except:

pass

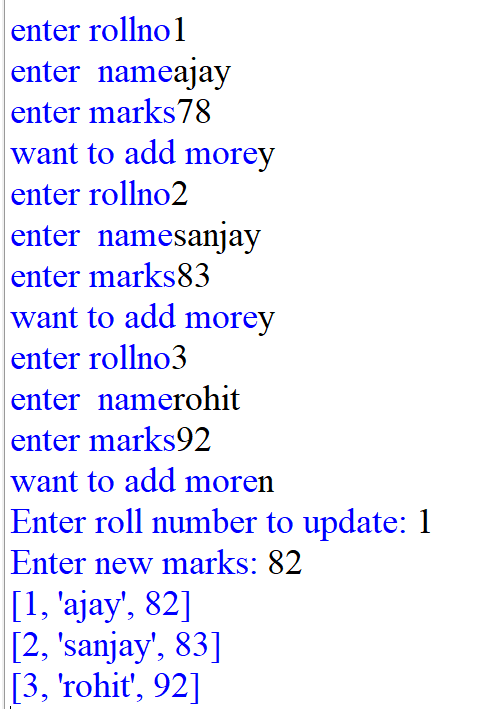
add()

roll = int(input("Enter roll number to update: "))

up(roll)

show()

OUTPUT



Q19 # Following is the structure of each record in a data file named ”PRODUCT.DAT”.

{"prod\_code":value, "prod\_desc":value, "stock":value}

The values for prod\_code and prod\_desc are strings, and the value for stock is an integer. Write a function in PYTHON to update the file with a new value of stock. The stock and the product\_code, whose stock is to be updated, are to be input during the execution of the function.

Code

import pickle

f=open('product.dat','wb')

while True:

a=int(input('enter product code'))

b=input('enter product description')

c=int(input('enter stock value'))

d={'prod\_code':a,'prod\_desc':b,'stock':c}

pickle.dump(d,f)

ch=input('do you wanna add more')

if ch=='n':

break

f.close()

def update\_stock(p,s):

updated = []

with open("PRODUCT.DAT", "rb") as f:

try:

while True:

item = pickle.load(f)

if item['prod\_code']==p:

item["stock:"]=s

updated.append(item)

except EOFError:

pass

with open("PRODUCT.DAT", "wb") as f:

for item in updated:

pickle.dump(item, f)

def display():

with open("PRODUCT.DAT", "rb") as f:

try:

while True:

item = pickle.load(f)

print(item)

except EOFError:

pass

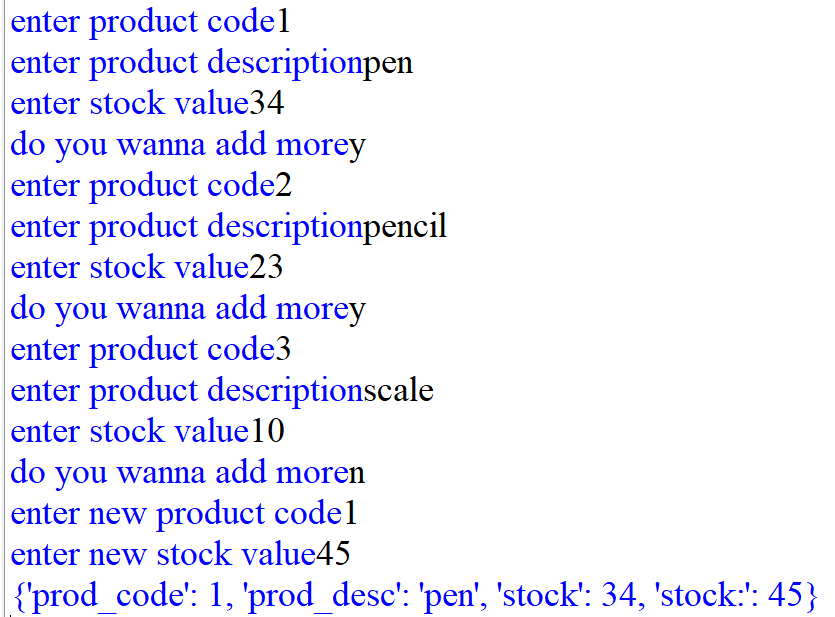
p=int(input('enter new product code'))

s=int(input('enter new stock value'))

update\_stock(p,s)

display()

OUTPUT



Q20 # Write a Python program to get 10 items’ details (itemno,name,price,category) from the user and create a CSV file(Items.csv).

Code

import csv

with open("Items.csv", "w", newline='') as f:

writer = csv.writer(f)

writer.writerow(["ItemNo", "Name", "Price", "Category"])

for \_ in range(10):

data = input("Enter itemno, name, price, category: ").split(',')

writer.writerow(data)

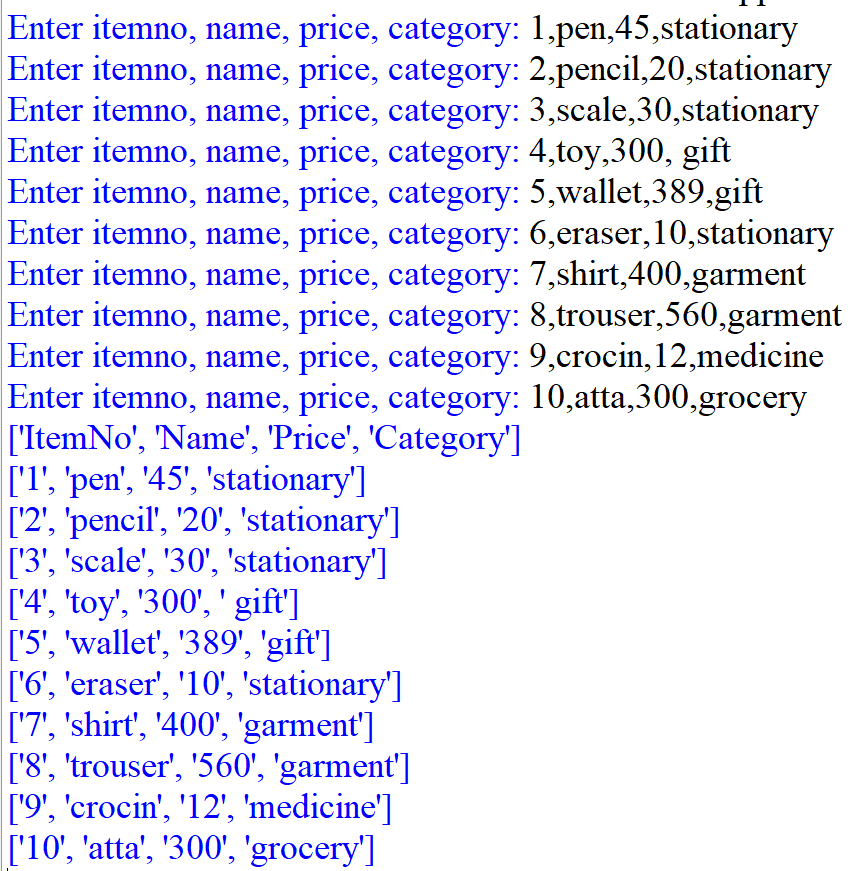
with open('Items.csv','r') as f1:

d=csv.reader(f1)

for i in d:

print(i)

OUTPUT



Q21 # Write a menu-based program to perform the operation on stack in python.

Code

stack = []

while True:

print("1.Push 2.Pop 3.Display 4.Exit")

ch = int(input("Enter choice: "))

if ch == 1:

stack.append(input("Enter item: "))

elif ch == 2:

if stack: print("Popped:", stack.pop())

else: print("Empty stack")

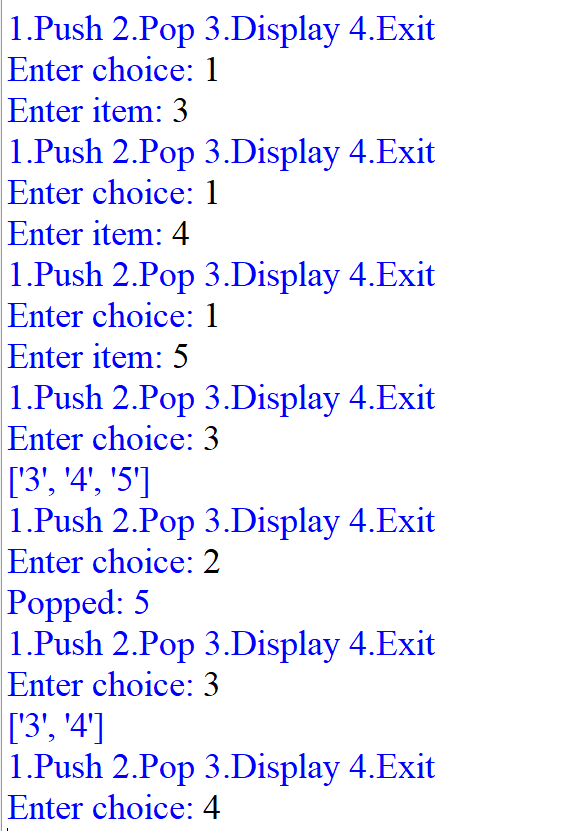
elif ch == 3:

print(stack)

else:

break

OUTPUT



Q22 # Write a program to connect Python with MySQL using database connectivity and perform the

following operations on data in database: Fetch, Update and delete the data

Code

import mysql.connector

con=mysql.connector.connect(host='localhost',passwd='1234',user='root',database='school')

cur=con.cursor()

#update

s=input('enter designation')

cur.execute("update emp set sal=sal+sal\*0.20 where designation='{}' ".format(s))

con.commit()

#fetch

cur.execute('select \* from emp')

r=cur.fetchone()

for i in r:

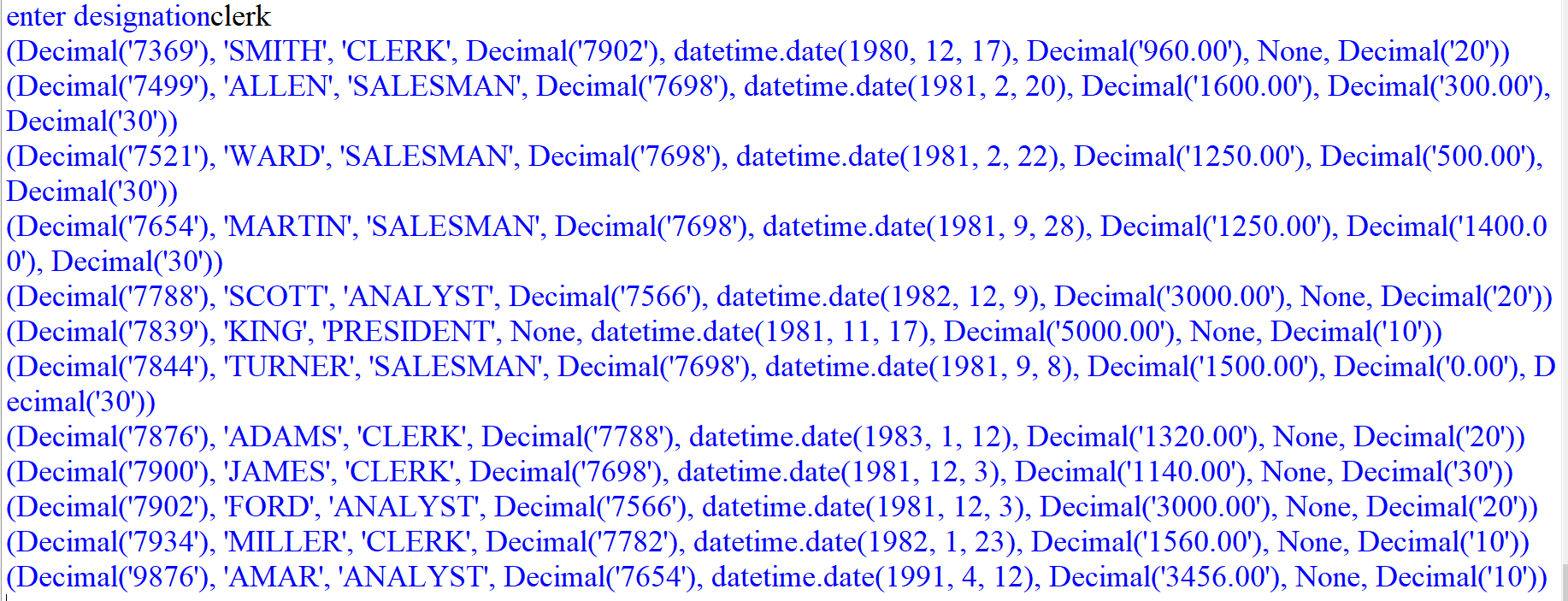
print(i)

z=cur.fetchmany(2)

for i in z:

print(i)

OUTPUT



Q23 **Consider the following tables STORE and SUPPLIERS and answer (a) and (b) parts of this question:**

**Table: - Store**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **ItemNo.** | **Item** | **Scode** | **Qty** | **Rate** | **LastBuy** |
| 2005 | Sharpener Classic | 23 | 60 | 8 | 31-Jun-09 |
| 2003 | Ball Pen 0.25 | 22 | 50 | 25 | 01-Feb-10 |
| 2002 | Gel Pen Premium | 21 | 150 | 12 | 24-Feb-10 |
| 2006 | Gel Pen Classic | 21 | 250 | 20 | 11-Mar-09 |
| 2001 | Eraser Small | 22 | 220 | 6 | 19-Jan-09 |
| 2004 | Eraser Big | 22 | 110 | 8 | 02-Dec-09 |
| 2009 | Ball Pen 0.5 | 21 | 180 | 18 | 03-Nov-09 |

**Table:- Supplier**

|  |  |
| --- | --- |
| **Scode** | **Sname** |
| 21 | Premium Stationers |
| 23 | Soft Plastics |
| 22 | Tetra Supply |

**A.Write SQL commands for the following statements:**

1. To display details of all the items in the Store table in ascending order of LastBuy.

Ans select \* from store order by LastBuy;

1. To display ItemNo and Item name of those items from Store table, whose Rate is more than 15 Rupees.

Ans select Itemno,Item from store where rate>15;

1. To display the details of those items whose Supplier code (Scode) is 22 or Quantity in Store (Qty) is more than 110 from the table Store.

Ans select \* from store where scode =22 or qty >110;

1. **To display Minimum Rate of items for each Supplier individually as per Scode from the table Store**

**Ans select min(rate) from store group by scode;**

**B.Give the output of the following SQL queries:**

1. SELECT COUNT(DISTINCT Scode) FROM Store;

|  |
| --- |
| COUNT(DISTINCT Scode) |
| 3 |

1. SELECT Rate\*Qty FROM Store WHERE ItemNo=2004;

|  |
| --- |
| Rate\*Qty |
| 880 |

1. SELECT Item,Sname FROM Store S, Suppliers P WHERE S.Scode=P.Scode AND ItemNo=2006;

|  |  |
| --- | --- |
| Item | Sname |
| Gel Pen Classic | Premium Stationers |

1. SELECT MAX(LastBuy) FROM Store;

|  |
| --- |
| MAX(LastBuy) |
| 24-Feb-10 |

Q24

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Consider the following table Item and Customer.**  **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 | Banglore | PC01 | |
| 1. **Write SQL commands for the following statements:** 2. To display the details of those Customer whose City is Delhi.   Ans select \* from customer where city=’Delhi’;   1. To display the details of Item whose Price is in the range of 3500 to 55000 (Both values included).   Ans select \* from item where price between 3500 and 55000   1. To display the CustomerName, City from table Customer, and ItemName and Price from table Item with their corresponding matching I\_ID.   Ans select customername,city,itemname,price from customer,item where item.I\_id=customer.I\_id;   1. To increase the Price of all Items by 1000 in the table Item.   Ans update item set price=price+1000;   1. **Give the output of the following SQL queries:** 2. SELECT DISTINCT City FROM Customer;  |  | | --- | | DISTINCT City | | Delhi | | Mumbai | | Banglore |  1. SELECT ItemName, Max(Price), Count(\*) FROM Item GROUP BY ItemName;  |  |  |  | | --- | --- | --- | | ItemName | Max(Price) | Count(\*) | | Personal Computer | 37000 | 3 | | Laptop | 57000 | 2 |  1. SELECT CustomerName, Manufacturer From Item, Customer WHERE Item.I\_Id=Customer.I\_Id;  |  |  | | --- | --- | | CustomerName | Manufacturer | | K Agarwal | ABC | | H Singh | XYZ | | R Pandey | COMP | | C Sharma | PQR |  1. SELECT ItemName, Price \* 100 FROM Item WHERE Manufacturer = ‘ABC’;  |  |  | | --- | --- | | ItemName | Price \* 100 | | Personal Computer | 3500000 | | Laptop | 5500000 | |

Q25 **Consider the following tables.**

**TABLE : SENDER**

|  |  |  |  |
| --- | --- | --- | --- |
| **SenderID** | **SenderName** | **SenderAddress** | **SenderCity** |
| ND01 | R Jain | 2,ABC Appts | New Delhi |
| MU02 | H Sinha | 12, Newtown | Mumbai |
| MU15 | S Jha | 27/A, Park Street | Mumbai |
| ND50 | T Prasad | 122-K, SDA | New Delhi |

**TABLE : RECIPIENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **RecID** | **SenderID** | **RecName** | **RecAddress** | **ReCity** |
| KO05 | ND01 | R Bajpayee | 5, Central Avenue | Kolkata |
| ND08 | MU02 | S Mohan | 116, A vihar | New Delhi |
| MU19 | ND01 | H singh | 2a, Andheri east | Mumbai |
| MU32 | MU15 | P K Swamy | B5, c S Terminus | Mumbai |
| ND48 | ND50 | S Tirupathi | 13, B1 d, Mayur vihar | New Delhi |

1. **Write SQL commands for the following statements:**
2. To display the names of all Senders from Mumbai.

Ans select sendername from sender where sendercity=’mumbai’;

1. To display the RecID, SenderName, SenderAddress, RecName, RecAddess for every Recipient

Ans select recid,sendername,senderaddress,recname,recaddress from sender,recipient where sender.senderid=recipient.senderid;

1. To display Recipient detail in ascending order of RecName

Ans select \* from recipient order by recname;

1. To display number of Recipients from each city

Ans seect count(\*) from recipient group by recity;

1. **Give the output of the following SQL queries:**
2. SELECT DISTINCT Sendercity FROM Sender;

|  |
| --- |
| DISTINCT Sendercity |
| New Delhi |
| Mumbai |

1. SELECT A.SenderName, B.RecName FROM Sender A, Recipient B WHERE A.SenderID=B.SenderID AND B.RecCity=’Mumbai’;

|  |  |
| --- | --- |
| SenderName | RecName |
| R Jain | H singh |
| S Jha | P K Swamy |

1. SELECT RecName, RecAddress FROM Recipient WHERE recCity NOT IN(‘Mumbai’, ‘Kolkata’);

|  |  |
| --- | --- |
| RecName | RecAddress |
| S Mohan | 116, A vihar |
| S Tirupathi | 13, B1 d, Mayur vihar |

1. SELECT RecID, RecName FROM Recipient WHERE SenderID=’MU02’ OR SenderID=’ND50’;

|  |  |
| --- | --- |
| RecID | RecName |
| ND08 | S Mohan |
| ND48 | S Tirupathi |

Q26 **Consider the following tables FURNITURE and ARRIVALS: write SQL commands for the questions (a) to (f) and give outputs for SQL queries (i) to (iv) :**

**TABLE : FURNITURE**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **ITEMNAME** | **TYPE** | **DATEOFSTOCK** | **PRICE** | **DISCOUNT** |
| 1 | White lotus | Double Bed | 23/02/02 | 30000 | 25 |
| 2 | Pink feather | Baby cot | 20/01/02 | 7000 | 20 |
| 3 | Dolphin | Baby cot | 19/02/02 | 9500 | 20 |
| 4 | Decent | Office Table | 01/01/02 | 25000 | 30 |
| 5 | Comfort zone | Double Bed | 12/01/02 | 25000 | 25 |
| 6 | Donald | Baby cot | 24/02/02 | 6500 | 15 |
| 7 | Royal Finish | Office Table | 20/02/02 | 18000 | 30 |
| 8 | Royal tiger | Sofa | 22/02/02 | 31000 | 30 |
| 9 | Econo sitting | Sofa | 13/12/01 | 9500 | 25 |
| 10 | Eating Paradise | Dining Table | 19/02/02 | 11500 | 25 |

**TABLE : ARRIVALS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **NO** | **ITEMNAME** | **TYPE** | **DATEOFSTOCK** | **PRICE** | **DISCOUNT** |
| 11 | Wood Comfort | Double Bed | 23/03/03 | 25000 | 25 |
| 12 | Old Fox | Sofa | 20/02/03 | 17000 | 20 |
| 13 | Micky | Baby cot | 21/02/03 | 7500 | 15 |

1. **Write SQL commands for the following statements:**
2. To show all information about the Baby cots from the FURNITURE table.

Ans select \* from furniture where type=’Baby cot’;

1. To list the ITEMNAME which are priced at more than 15000 from the FURNITURE table.

Ans select itemname from furniture where price >15000;

1. To list ITEMNAME and TYPE of those items, in which DATEOFSTOCK is before 22/01/02 from the FURNITURE table in descending order of ITEMNAME.

Ans select itemname,type from furniture where dateofstock<’22/01/02’ order by itemname desc;

1. To display ITEMNAME and DATEOFSTOCK of those items, in which the DISCOUNT percentage is more than 25 form FURNITURE table.

Ans select itemname,dateofstock from furniture where discount>25;

1. To count the number of items, whose TYPE is “Sofa” from FURNITURE table.

Ans select count(\*) from furniture where type=’sofa’;

1. To insert a new row in the ARRIVALS table with the following data:

14, ‘Velvet touch’, ‘Double bed’, {25/03/03}, 25000,30

Ans insert into arrivals values (14, ‘Velvet touch’, ‘Double bed’, ‘25/03/03’, 25000,30);

1. **Give the output of following SQL statement:**

*NOTE : Outputs of the below mentioned queries should be based on original data given in both the tables, i.e., without considering the insertion done in (f) part of this question :*

I . Select COUNT (distinct TYPE) from FURNITURE;

|  |
| --- |
| COUNT (distinct TYPE) |
| 6 |

1. Select MAX(DISCOUNT) form FURNITURE;

|  |
| --- |
| MAX(DISCOUNT) |
| 30 |

1. Select AVG(DISCOUNT) form FURNITURE where TYPE = ‘Baby cot’;

|  |
| --- |
| AVG(DISCOUNT) |
| 7666.6666 |

1. Select SUM(PRICE) from FURNITURE where DATEOFSTOCK<{12/02/02};

|  |
| --- |
| sum(price) |
| 66500 |

Q27

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Study the following tables DOCTOR and SALARY and write SQL commands for the questions (i) to (iv) and give outputs for SQL queries (v) to (vi) :**  **TABLE : DOCTOR**   |  |  |  |  |  | | --- | --- | --- | --- | --- | | **ID** | **NAME** | **DEPT** | **SEX** | **EXPERIENCE** | | 101 | John | ENT | M | 12 | | 104 | Smith | ORTHOPEDIC | M | 5 | | 107 | George | CARDIOLOGY | M | 10 | | 114 | Lara | SKIN | F | 3 | | 109 | K George | MEDICINE | F | 9 | | 105 | Johnson | ORTHOPEDIC | M | 10 | | 117 | Lucy | ENT | F | 3 | | 111 | Bill | MEDICINE | F | 12 | | 130 | Morphy | ORTHOPEDIC | M | 15 | |
| **TABLE : SALARY**   |  |  |  |  | | --- | --- | --- | --- | | **ID** | **BASIC** | **ALLOWANCE** | **CONSULTATION** | | 101 | 12000 | 1000 | 300 | | 104 | 23000 | 2300 | 500 | | 107 | 32000 | 4000 | 500 | | 114 | 12000 | 5200 | 100 | | 109 | 42000 | 1700 | 200 | | 105 | 18900 | 1690 | 300 | | 130 | 21700 | 2600 | 300 | |
| 1. **Write SQL commands for the following statements:**    * 1. Display NAME of all doctors who are in “MEDICINE” having more than 10 year experience from the table DOCTOR.   Ans select name from doctor where dept=’medicine’ and experience>10;   * + 1. Display the average salary of all doctors working in “ENT” department using the tables DOCTOR and SALARY. Salary=BASIC + ALLOWANCE.   Ans select avg(basic+allowance) from doctor,salary where doctor.id=salary.id;   * + 1. Display the minimum ALLOWANCE of female doctors.   Ans select min(allownance) from salary,doctor where doctor.id=salary.id and sex=’f’;   * + 1. Display the highest consultation fee among all male doctor   Ans select max(consultation) from salary,doctor where doctor.id=salary.id and sex=’m’; |
| 1. **Give the output of following SQL statement:**    * 1. SELECT count(\*) from DOCTOR where SEX=”F”.  |  | | --- | | Count(\*) | | 4 |  * + 1. SELECT NAME, DEPT, BASIC from DOCTOR Salary WHERE DEPT=”ENT” AND DOCTORID=SALARY.ID  |  |  |  | | --- | --- | --- | | Name | Dept | Basic | | John | ENT | 12000 | |