Computer Science

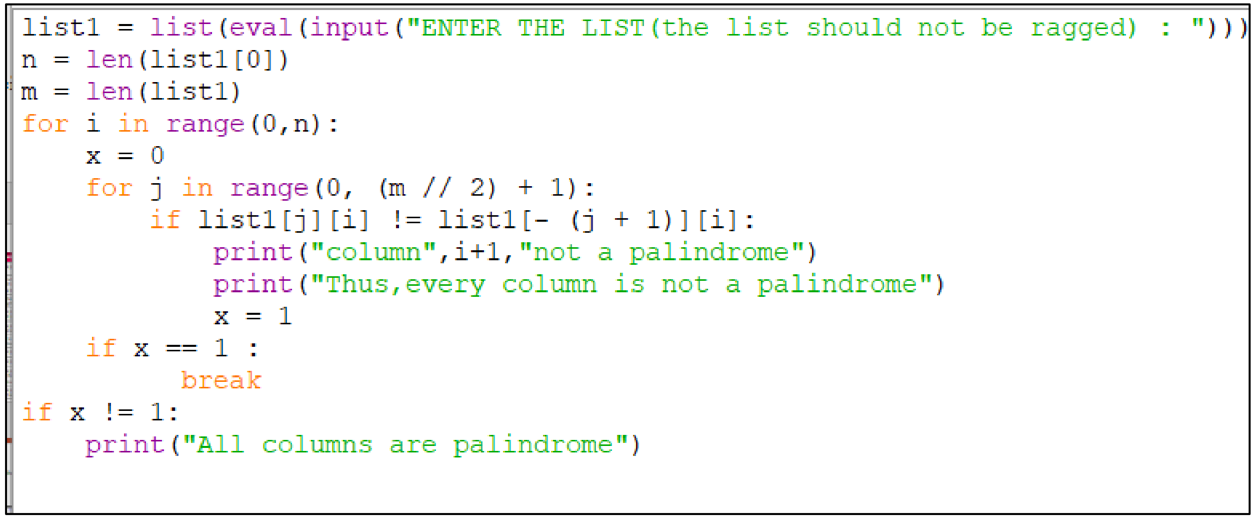
Practical File

Riya Verma , XII B

Program 1

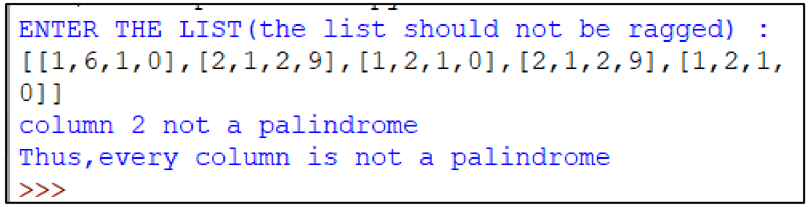
Write a program to accept a 2-D array from the user and check whether every column is a palindrome or not

INPUT

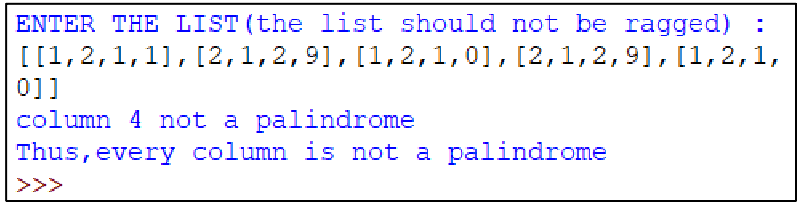


```
list1 = list(eval(input("ENTER THE LIST(the list should not be ragged) : ")))
n = len(list1[0])
m = len(list1)
for i in range(0,n):
    x = 0
    for j in range(0, (m // 2) + 1):
        if list1[j][i] != list1[- (j + 1)][i]:
            print("column",i+1,"not a palindrome")
            print("Thus,every column is not a palindrome")
            x = 1
    if x == 1 :
          break
if x != 1:
    print("All columns are palindrome")
```

OUTPUT



```
ENTER THE LIST(the list should not be ragged) :
[[1,6,1,0],[2,1,2,9],[1,2,1,0],[2,1,2,9],[1,2,1,
0]]
column 2 not a palindrome
Thus,every column is not a palindrome
>>>
```



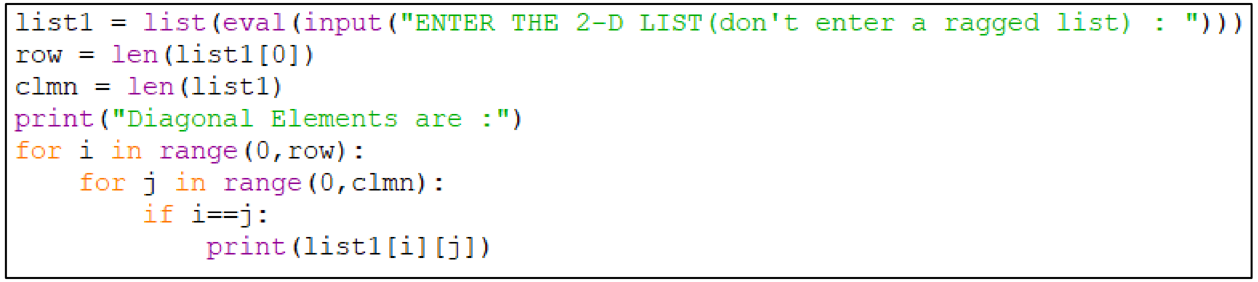
```
ENTER THE LIST(the list should not be ragged) :
[[1,2,1,1],[2,1,2,9],[1,2,1,0],[2,1,2,9],[1,2,1,
0]]
column 4 not a palindrome
Thus,every column is not a palindrome
>>>
```

Riya Verma , XII B

Program 2

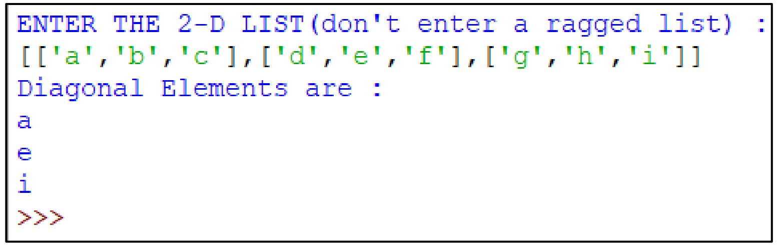
Write a program to accept a 2-D array from the user and print its diagonal elements

INPUT



```
list1 = list(eval(input("ENTER THE 2-D LIST(don't enter a ragged list) : ")))
row = len(list1[0])
clmn = len(list1)
print("Diagonal Elements are :")
for i in range(0,row):
    for j in range(0,clmn):
        if i==j:
            print(list1[i][j])
```

OUTPUT



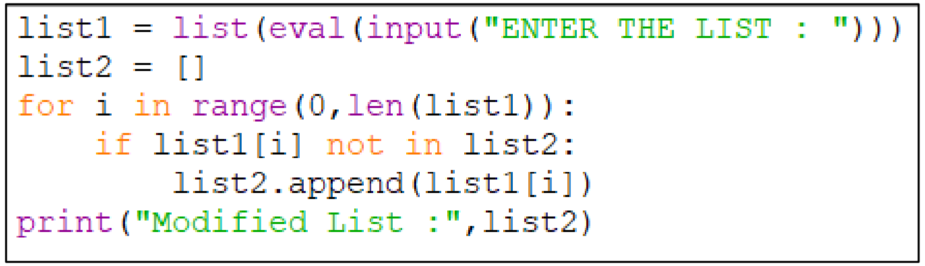
```
ENTER THE 2-D LIST(don't enter a ragged list) :
[['a','b','c'],['d','e','f'],['g','h','i']]
Diagonal Elements are :
a
e
i
>>>
```

Riya Verma , XII B

Program 3

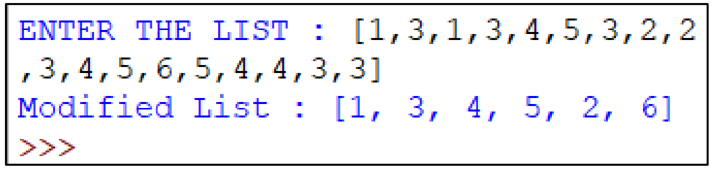
Write a program to remove all duplicate elements from the list

INPUT



```
list1 = list(eval(input("ENTER THE LIST : ")))
list2 = []
for i in range(0,len(list1)):
    if list1[i] not in list2:
        list2.append(list1[i])
print("Modified List :",list2)
```

OUTPUT

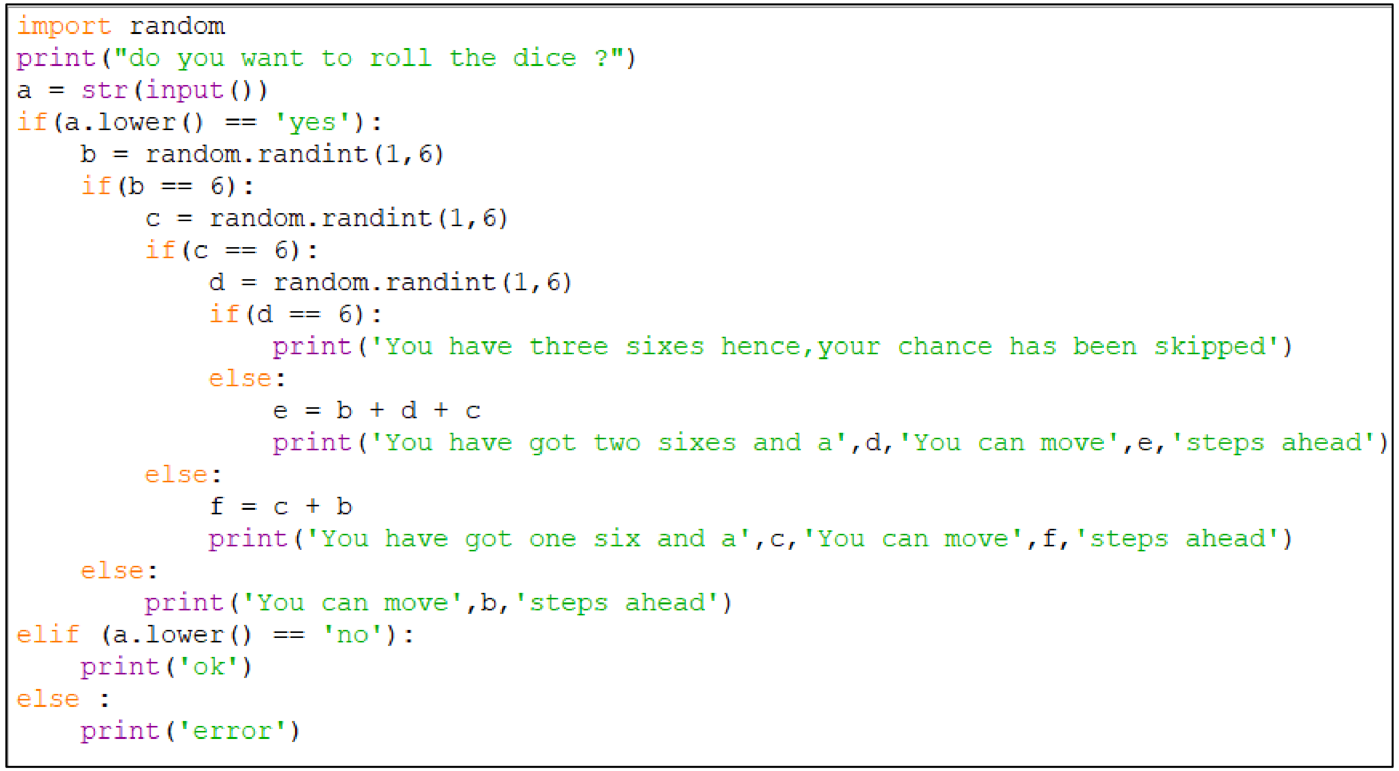


```
ENTER THE LIST : [1,3,1,3,4,5,3,2,2
,3,4,5,6,5,4,4,3,3]
Modified List : [1, 3, 4, 5, 2, 6]
>>>
```

Program 4

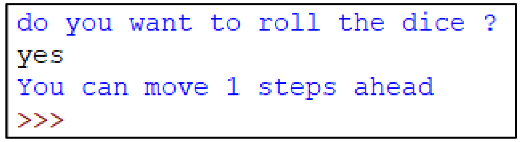
Write a program to implement random number generator that generates random numbers between 1 and 6 (simulates a dice)

INPUT

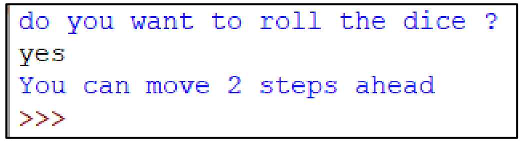


```
import random
print("do you want to roll the dice ?")
a = str(input())
if(a.lower() == 'yes'):
    b = random.randint(1,6)
    if(b == 6):
        c = random.randint(1,6)
        if(c == 6):
            d = random.randint(1,6)
            if(d == 6):
                print('You have three sixes hence,your chance has been skipped')
            else:
                e = b + d + c
                print('You have got two sixes and a',d,'You can move',e,'steps ahead')
        else:
            f = c + b
            print('You have got one six and a',c,'You can move',f,'steps ahead')
    else:
        print('You can move',b,'steps ahead')
elif (a.lower() == 'no'):
    print('ok')
else :
    print('error')
```

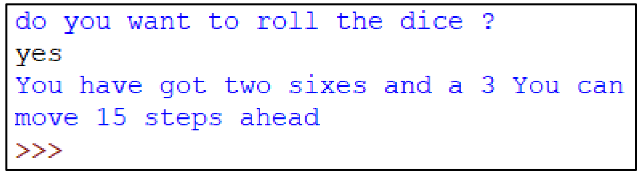
OUTPUT



```
do you want to roll the dice ?
yes
You can move 1 steps ahead
>>>
```



```
do you want to roll the dice ?
yes
You can move 2 steps ahead
>>>
```



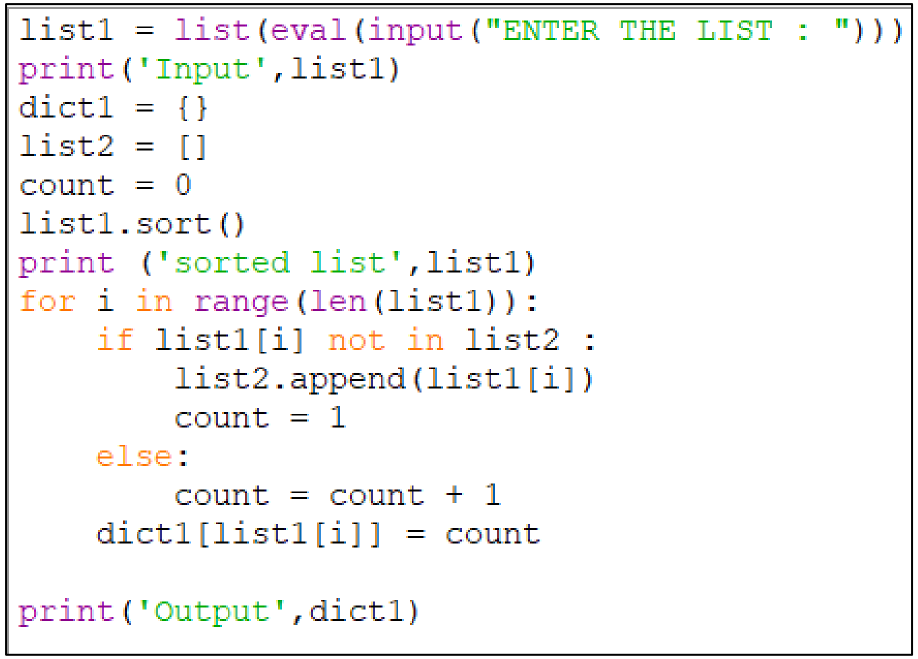
```
do you want to roll the dice ?
yes
You have got two sixes and a 3 You can
move 15 steps ahead
>>>
```

Riya Verma , XII B

Program 5

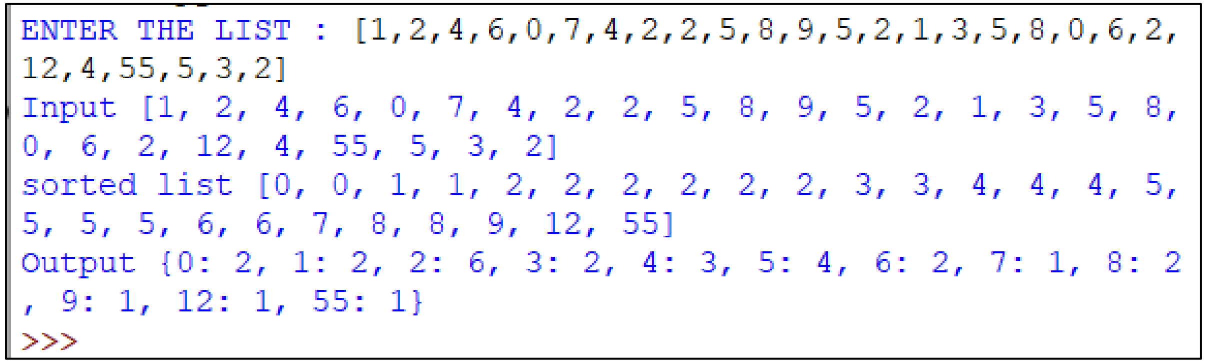
Write a program to accept a list from the user to count and store the frequency of each of its element in a dictionary. Also, sort the dictionary by its keys

INPUT



```
list1 = list(eval(input("ENTER THE LIST : ")))
print('Input',list1)
dict1 = {}
list2 = []
count = 0
list1.sort()
print ('sorted list',list1)
for i in range(len(list1)):
    if list1[i] not in list2 :
        list2.append(list1[i])
        count = 1
    else:
        count = count + 1
    dict1[list1[i]] = count

print('Output',dict1)
```

OUTPUT



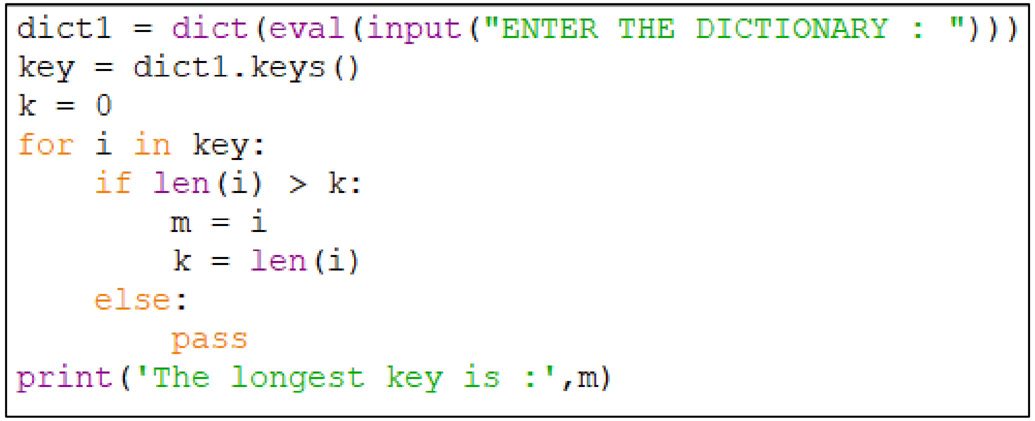
```
ENTER THE LIST : [1,2,4,6,0,7,4,2,2,5,8,9,5,2,1,3,5,8,0,6,2,
12,4,55,5,3,2]
Input [1, 2, 4, 6, 0, 7, 4, 2, 2, 5, 8, 9, 5, 2, 1, 3, 5, 8,
0, 6, 2, 12, 4, 55, 5, 3, 2]
sorted list [0, 0, 1, 1, 2, 2, 2, 2, 2, 2, 3, 3, 4, 4, 4, 5,
5, 5, 5, 6, 6, 7, 8, 8, 9, 12, 55]
Output {0: 2, 1: 2, 2: 6, 3: 2, 4: 3, 5: 4, 6: 2, 7: 1, 8: 2
, 9: 1, 12: 1, 55: 1}
>>>
```

Riya Verma , XII B

Program 6

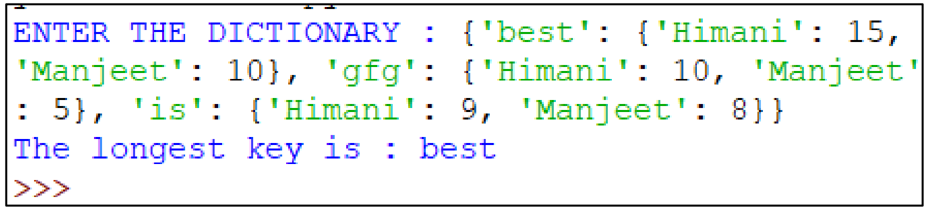
Write a program to accept a dictionary from the user. Find and display the key with maximum key length.

INPUT



```
dict1 = dict(eval(input("ENTER THE DICTIONARY : ")))
key = dict1.keys()
k = 0
for i in key:
    if len(i) > k:
        m = i
        k = len(i)
    else:
        pass
print('The longest key is :',m)
```

OUTPUT



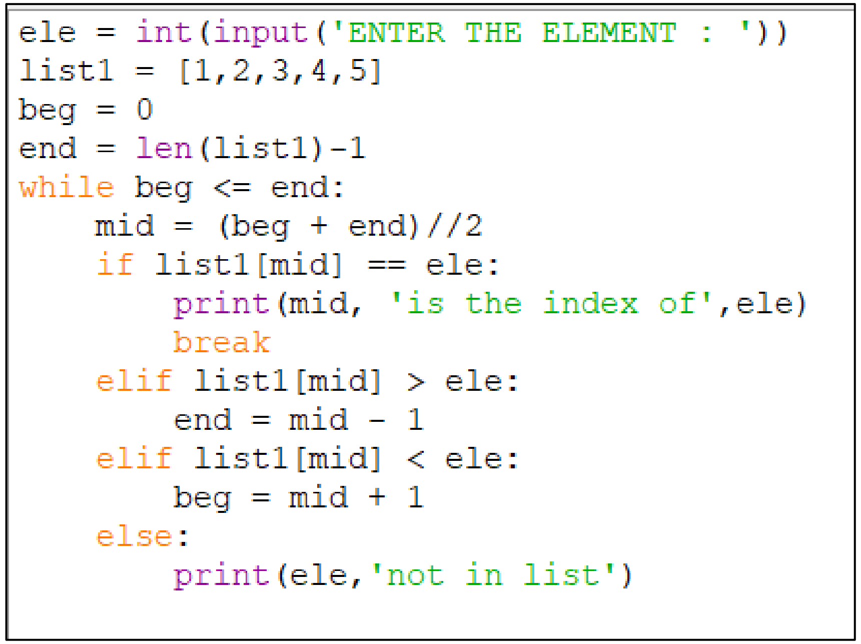
```
ENTER THE DICTIONARY : {'best': {'Himani': 15,
'Manjeet': 10}, 'gfg': {'Himani': 10, 'Manjeet'
: 5}, 'is': {'Himani': 9, 'Manjeet': 8}}
The longest key is : best
>>>
```

Riya Verma , XII B

Program 7

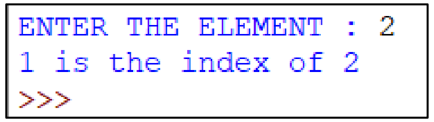
Write a program to search an element in a list using binary search algorithm.

INPUT

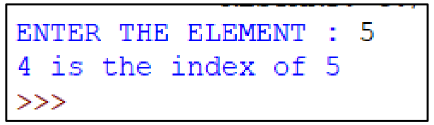


```
ele = int(input('ENTER THE ELEMENT : '))
list1 = [1,2,3,4,5]
beg = 0
end = len(list1)-1
while beg <= end:
    mid = (beg + end)//2
    if list1[mid] == ele:
        print(mid, 'is the index of',ele)
        break
    elif list1[mid] > ele:
        end = mid - 1
    elif list1[mid] < ele:
        beg = mid + 1
    else:
        print(ele,'not in list')
```

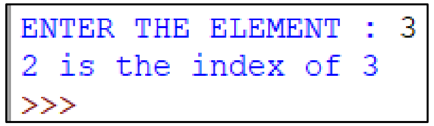
OUTPUT



```
ENTER THE ELEMENT : 2
1 is the index of 2
>>>
```



```
ENTER THE ELEMENT : 5
4 is the index of 5
>>>
```



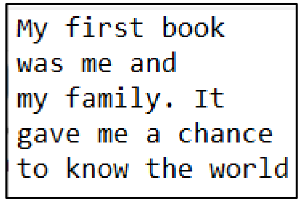
```
ENTER THE ELEMENT : 3
2 is the index of 3
>>>
```

Program 8

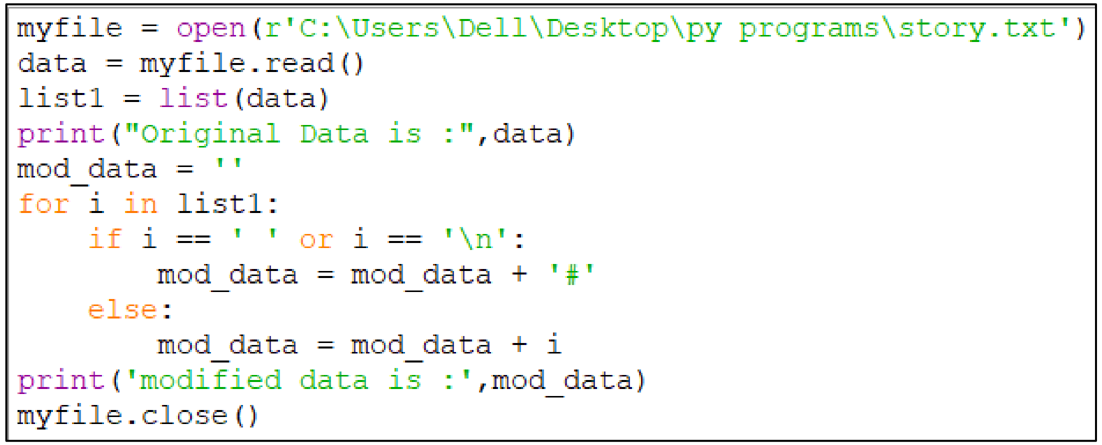
Write a program to read a text file “story.txt” line by line and display each word separated by a #.

INPUT

story.txt

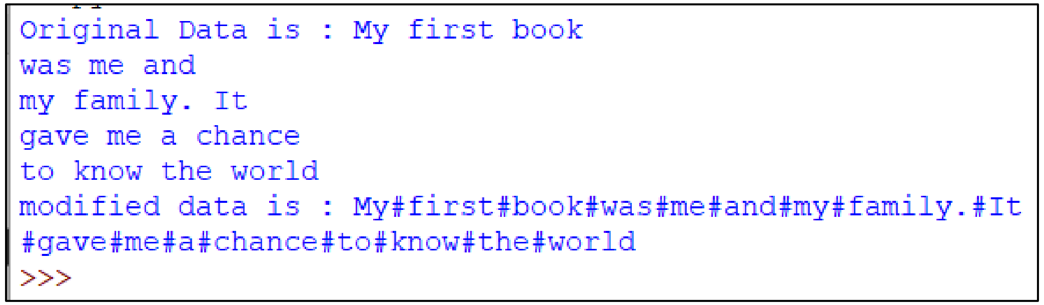


```
My first book
was me and
my family. It
gave me a chance
to know the world
```



```python
myfile = open(r'C:\Users\Dell\Desktop\py programs\story.txt')
data = myfile.read()
list1 = list(data)
print("Original Data is :",data)
mod_data = ''
for i in list1:
    if i == ' ' or i == '\n':
        mod_data = mod_data + '#'
    else:
        mod_data = mod_data + i
print('modified data is :',mod_data)
myfile.close()
```

OUTPUT



```
Original Data is : My first book
was me and
my family. It
gave me a chance
to know the world
modified data is : My#first#book#was#me#and#my#family.#It
#gave#me#a#chance#to#know#the#world
>>>
```

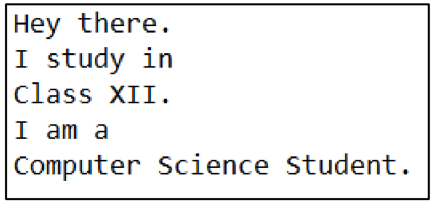
Riya Verma , XII B

Program 9

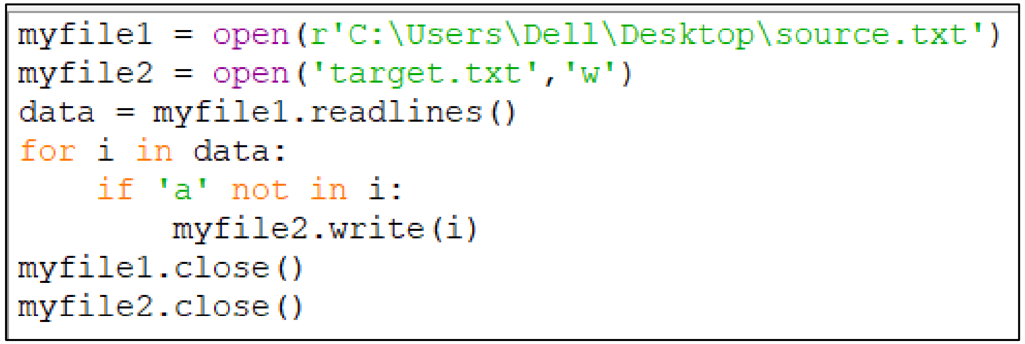
Write a program to remove all the lines that contain the character 'a' in a file “source.txt” and write it to another text file “target.txt”.

INPUT

source.txt

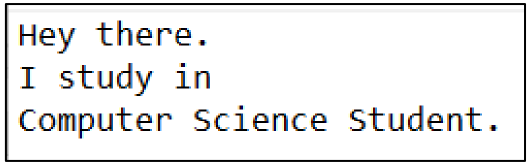


```
Hey there.
I study in
Class XII.
I am a
Computer Science Student.
```



```python
myfile1 = open(r'C:\Users\Dell\Desktop\source.txt')
myfile2 = open('target.txt','w')
data = myfile1.readlines()
for i in data:
    if 'a' not in i:
        myfile2.write(i)
myfile1.close()
myfile2.close()
```

OUTPUT



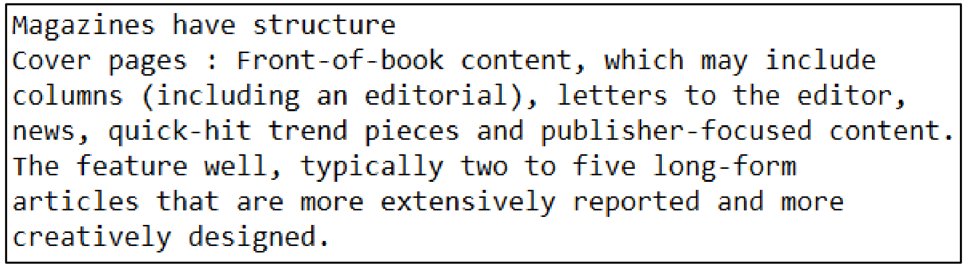
```
Hey there.
I study in
Computer Science Student.
```

Program 10

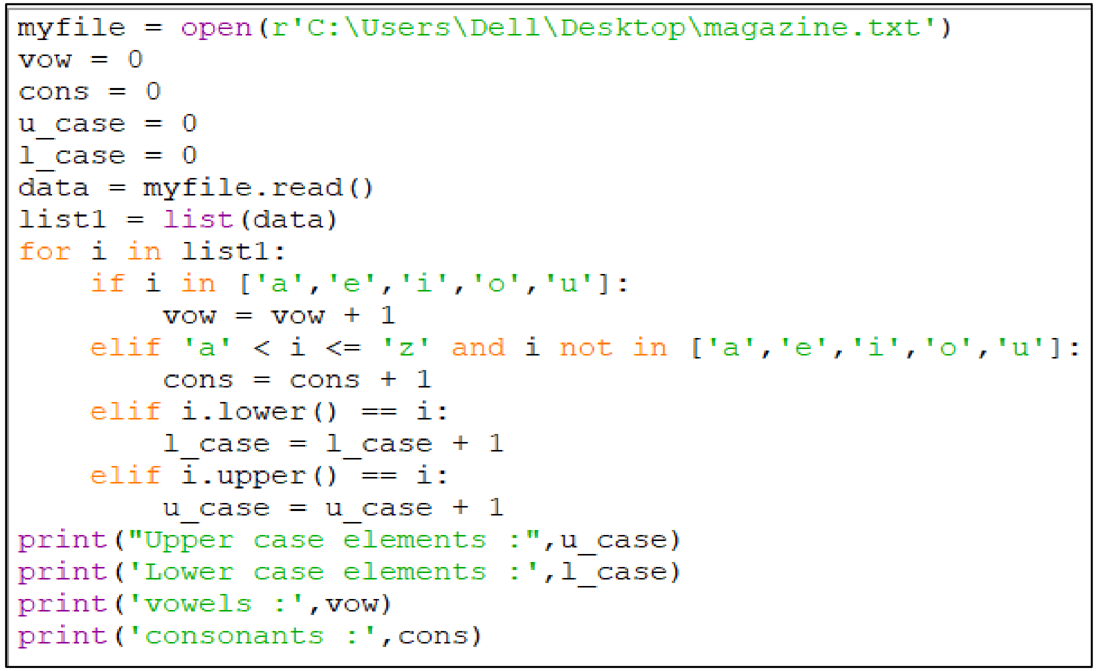
Write a program to read a text file “magazine.txt” and display the number of vowels/consonants/uppercase characters and lowercase characters.

magazine.txt

INPUT

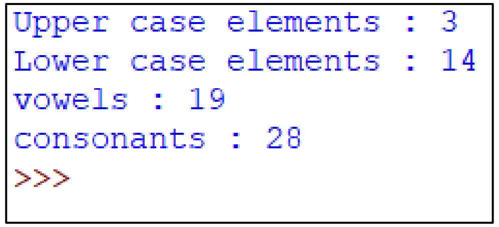


```
Magazines have structure
Cover pages : Front-of-book content, which may include
columns (including an editorial), letters to the editor,
news, quick-hit trend pieces and publisher-focused content.
The feature well, typically two to five long-form
articles that are more extensively reported and more
creatively designed.
```



```python
myfile = open(r'C:\Users\Dell\Desktop\magazine.txt')
vow = 0
cons = 0
u_case = 0
l_case = 0
data = myfile.read()
list1 = list(data)
for i in list1:
    if i in ['a','e','i','o','u']:
        vow = vow + 1
    elif 'a' < i <= 'z' and i not in ['a','e','i','o','u']:
        cons = cons + 1
    elif i.lower() == i:
        l_case = l_case + 1
    elif i.upper() == i:
        u_case = u_case + 1
print("Upper case elements :",u_case)
print('Lower case elements :',l_case)
print('vowels :',vow)
print('consonants :',cons)
```

OUTPUT



```
Upper case elements : 3
Lower case elements : 14
vowels : 19
consonants : 28
>>>
```

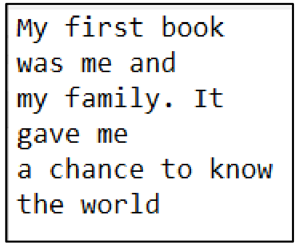
Riya Verma , XII B

Program 11

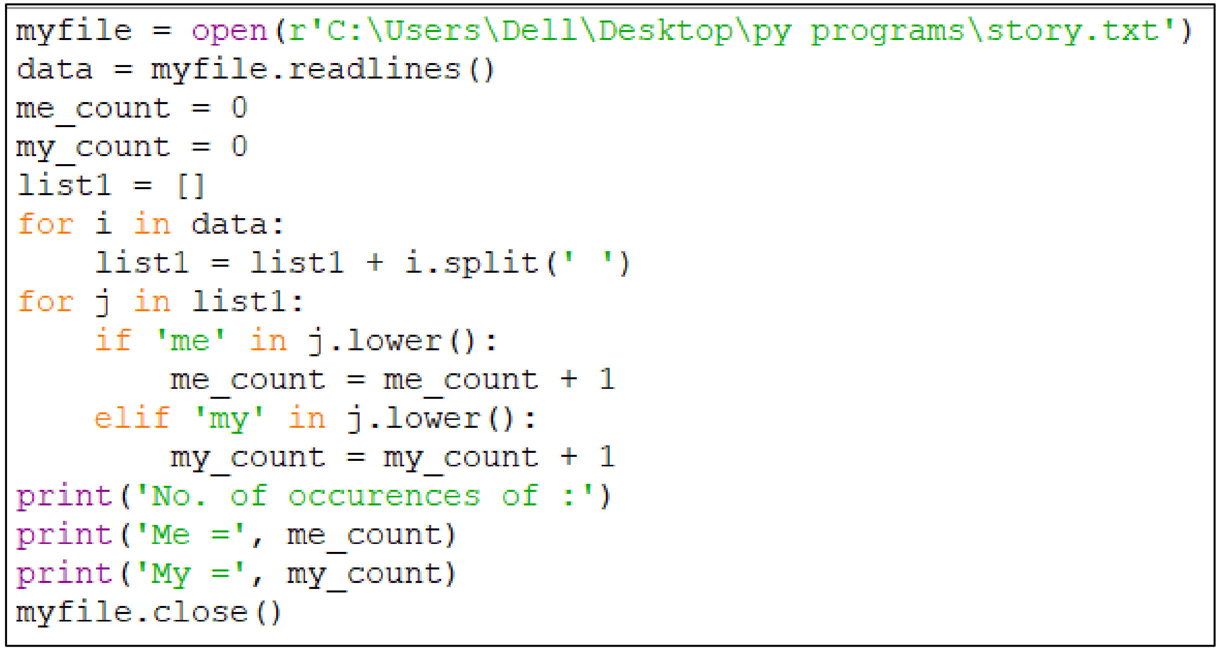
Write a program in python that counts the number of ‘Me’ or ‘My’ words present in the text file ‘story.txt’.

INPUT

story.txt

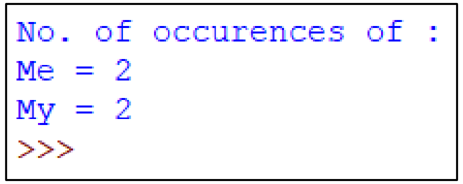


```
My first book
was me and
my family. It
gave me
a chance to know
the world
```



```
myfile = open(r'C:\Users\Dell\Desktop\py programs\story.txt')
data = myfile.readlines()
me_count = 0
my_count = 0
list1 = []
for i in data:
    list1 = list1 + i.split(' ')
for j in list1:
    if 'me' in j.lower():
        me_count = me_count + 1
    elif 'my' in j.lower():
        my_count = my_count + 1
print('No. of occurences of :')
print('Me =', me_count)
print('My =', my_count)
myfile.close()
```

OUTPUT



```
No. of occurences of :
Me = 2
My = 2
>>>
```

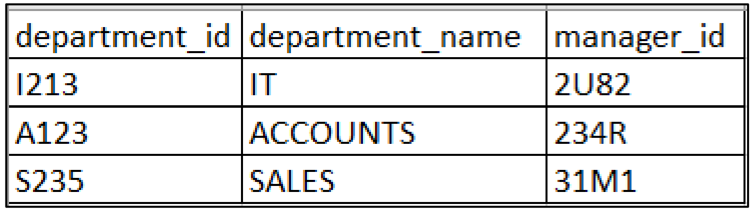
Riya Verma , XII B

Program 12

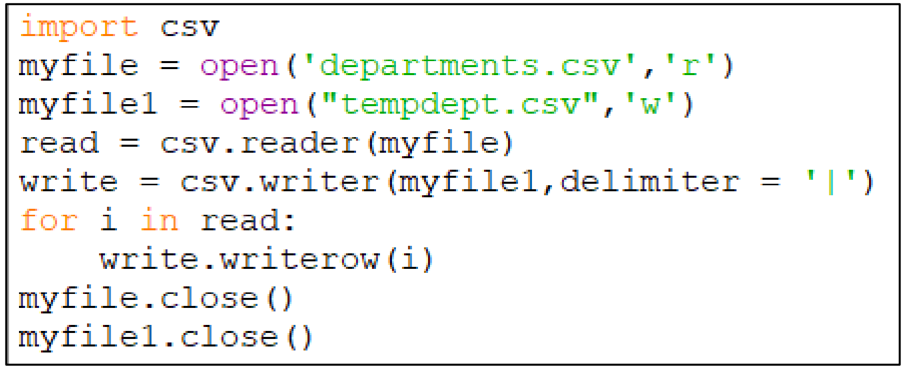
Write a Python program to read a given CSV file ‘departments.csv’ (department_id, department_name, manager_id) and write records into another CSV ‘tempdept.csv’ file with a different delimiter. Change delimiter from ‘,’ to ‘|’.

INPUT

departments.csv

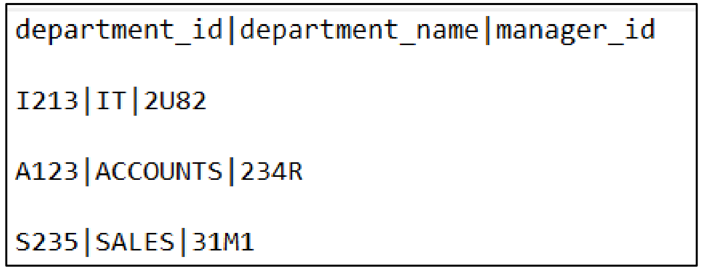


department_id	department_name	manager_id
I213	IT	2U82
A123	ACCOUNTS	234R
S235	SALES	31M1



```
import csv
myfile = open('departments.csv','r')
myfile1 = open("tempdept.csv",'w')
read = csv.reader(myfile)
write = csv.writer(myfile1,delimiter = '|')
for i in read:
    write.writerow(i)
myfile.close()
myfile1.close()
```

OUTPUT



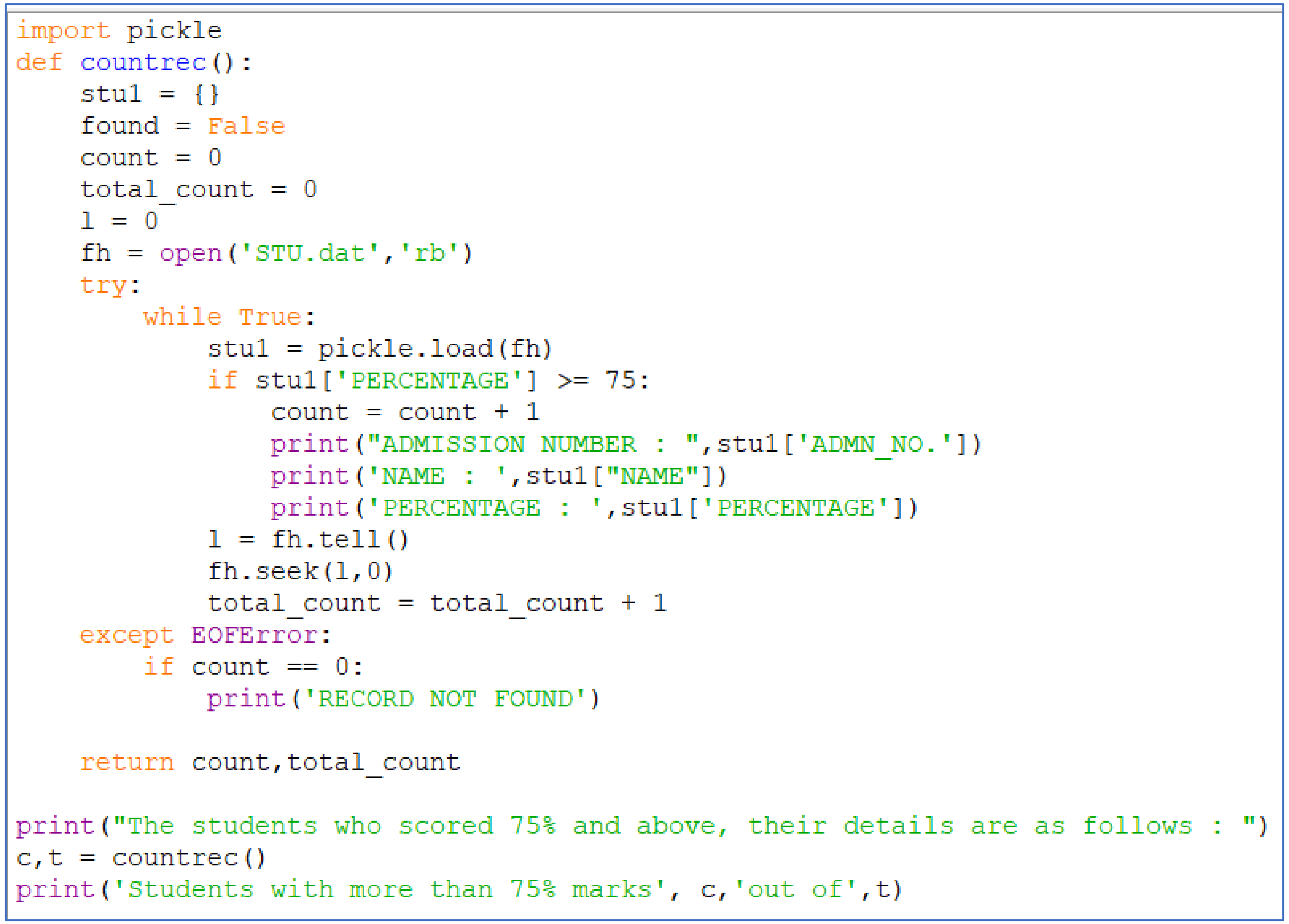
```
department_id|department_name|manager_id

I213|IT|2U82

A123|ACCOUNTS|234R

S235|SALES|31M1
```

Riya Verma , XII B

Program 13

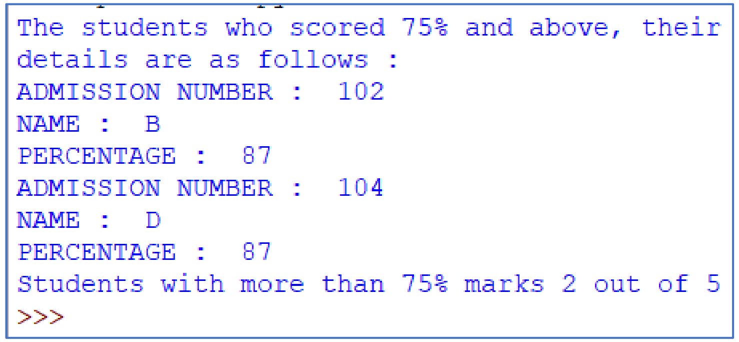
A binary file “STUDENT.DAT” has structure (admission_number, Name, Percentage). Write a program using function countrec() that would read contents of the file “STUDENT.DAT” and display the details of those students whose percentage is above 75. Also, display number of students scoring above 75%.

INPUT



```
import pickle
def countrec():
    stu1 = {}
    found = False
    count = 0
    total_count = 0
    l = 0
    fh = open('STU.dat','rb')
    try:
        while True:
            stu1 = pickle.load(fh)
            if stu1['PERCENTAGE'] >= 75:
                count = count + 1
                print("ADMISSION NUMBER : ",stu1['ADMN_NO.'])
                print('NAME : ',stu1["NAME"])
                print('PERCENTAGE : ',stu1['PERCENTAGE'])
            l = fh.tell()
            fh.seek(l,0)
            total_count = total_count + 1
    except EOFError:
        if count == 0:
            print('RECORD NOT FOUND')

    return count,total_count

print("The students who scored 75% and above, their details are as follows : ")
c,t = countrec()
print('Students with more than 75% marks', c,'out of',t)
```

OUTPUT



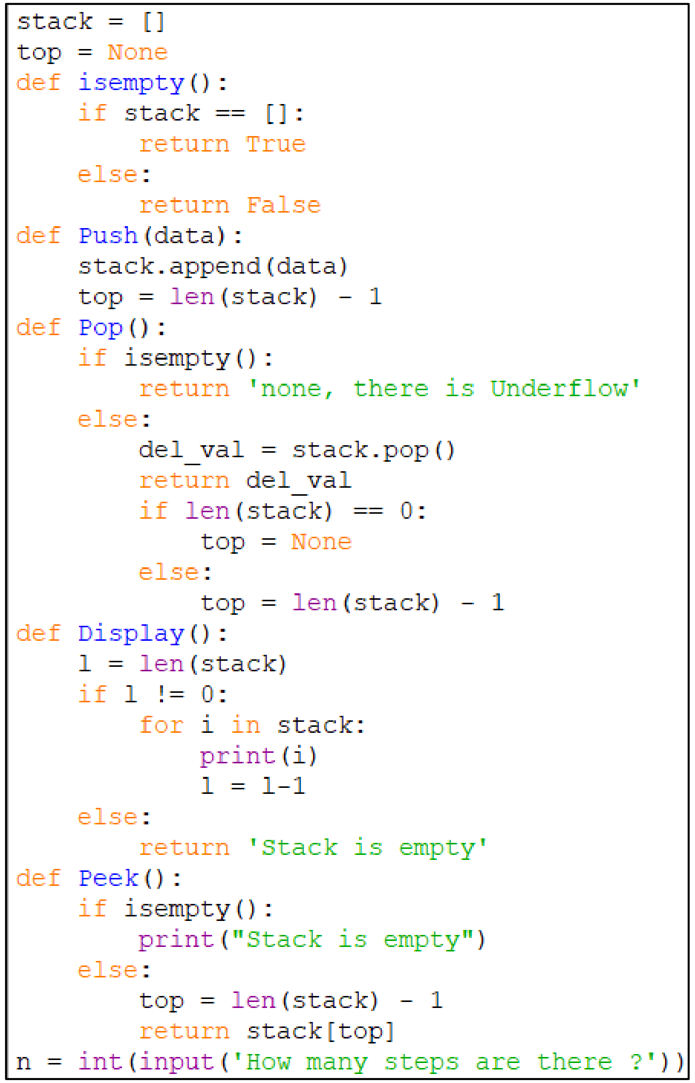
```
The students who scored 75% and above, their
details are as follows :
ADMISSION NUMBER :  102
NAME :  B
PERCENTAGE :  87
ADMISSION NUMBER :  104
NAME :  D
PERCENTAGE :  87
Students with more than 75% marks 2 out of 5
>>>
```

Riya Verma , XII B

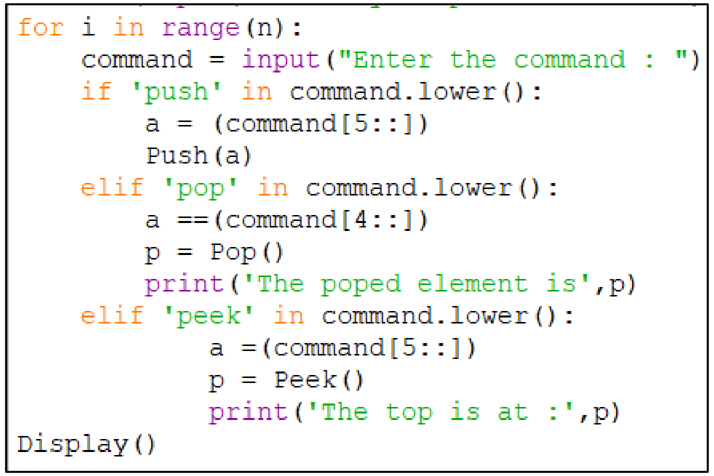
Program 14

Write a Python program to implement push, pop, peek and display operation in stack using list.

INPUT

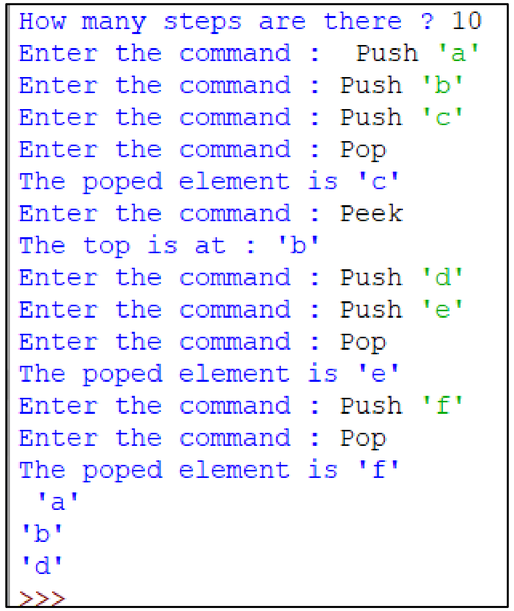


```
stack = []
top = None
def isempty():
    if stack == []:
        return True
    else:
        return False
def Push(data):
    stack.append(data)
    top = len(stack) - 1
def Pop():
    if isempty():
        return 'none, there is Underflow'
    else:
        del_val = stack.pop()
        return del_val
        if len(stack) == 0:
            top = None
        else:
            top = len(stack) - 1
def Display():
    l = len(stack)
    if l != 0:
        for i in stack:
            print(i)
            l = l-1
    else:
        return 'Stack is empty'
def Peek():
    if isempty():
        print("Stack is empty")
    else:
        top = len(stack) - 1
        return stack[top]
n = int(input('How many steps are there ?'))
```

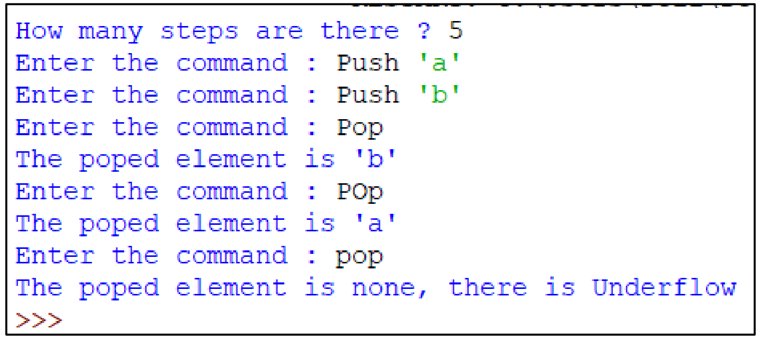


```
for i in range(n):
    command = input("Enter the command : ")
    if 'push' in command.lower():
        a = (command[5::])
        Push(a)
    elif 'pop' in command.lower():
        a ==(command[4::])
        p = Pop()
        print('The poped element is',p)
    elif 'peek' in command.lower():
        a =(command[5::])
        p = Peek()
        print('The top is at :',p)
Display()
```

OUTPUT



```
How many steps are there ? 10
Enter the command :  Push 'a'
Enter the command : Push 'b'
Enter the command : Push 'c'
Enter the command : Pop
The poped element is 'c'
Enter the command : Peek
The top is at : 'b'
Enter the command : Push 'd'
Enter the command : Push 'e'
Enter the command : Pop
The poped element is 'e'
Enter the command : Push 'f'
Enter the command : Pop
The poped element is 'f'
 'a'
'b'
'd'
>>>
```



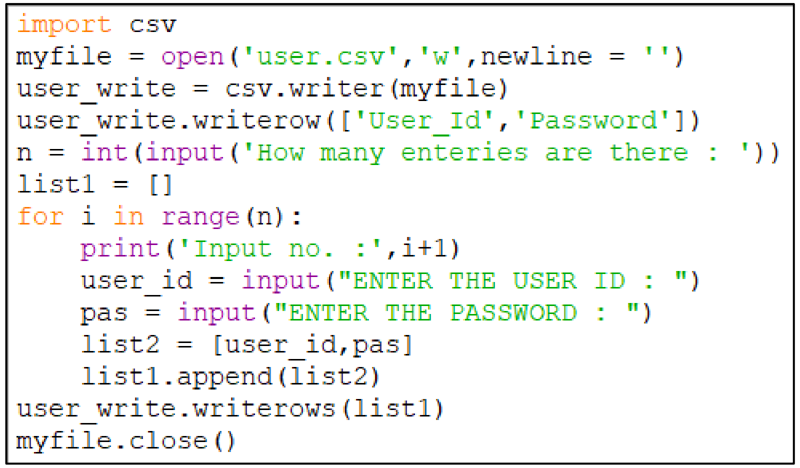
```
How many steps are there ? 5
Enter the command : Push 'a'
Enter the command : Push 'b'
Enter the command : Pop
The poped element is 'b'
Enter the command : POp
The poped element is 'a'
Enter the command : pop
The poped element is none, there is Underflow
>>>
```

Riya Verma , XII B

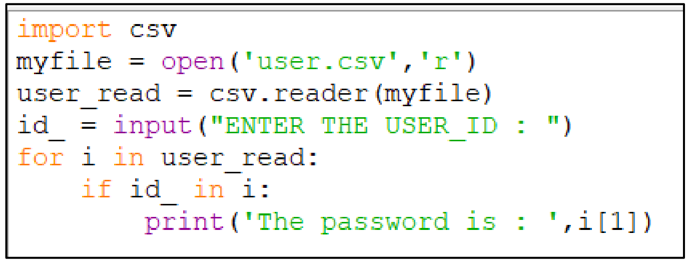
Program 15

Write a program to create a CSV file by entering user-id and password, read and search the password for given userid.

INPUT



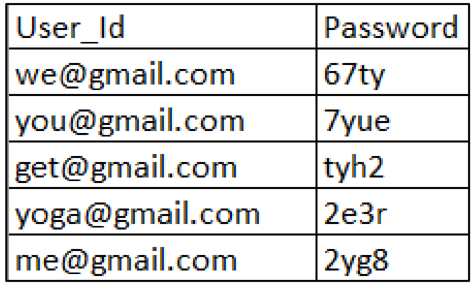
```python
import csv
myfile = open('user.csv','w',newline = '')
user_write = csv.writer(myfile)
user_write.writerow(['User_Id','Password'])
n = int(input('How many enteries are there : '))
list1 = []
for i in range(n):
    print('Input no. :',i+1)
    user_id = input("ENTER THE USER ID : ")
    pas = input("ENTER THE PASSWORD : ")
    list2 = [user_id,pas]
    list1.append(list2)
user_write.writerows(list1)
myfile.close()
```



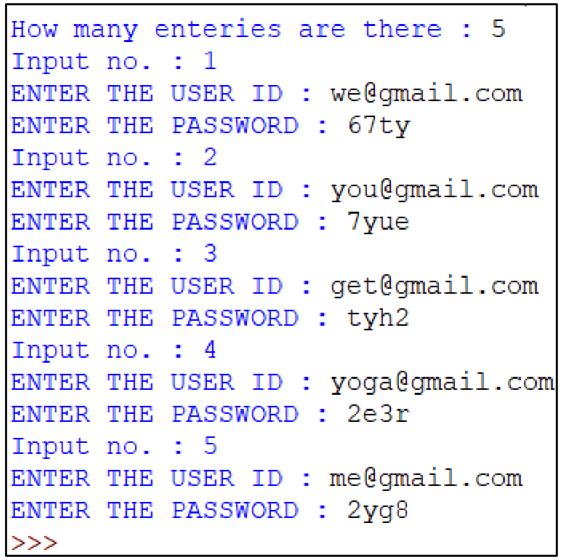
```python
import csv
myfile = open('user.csv','r')
user_read = csv.reader(myfile)
id_ = input("ENTER THE USER_ID : ")
for i in user_read:
    if id_ in i:
        print('The password is : ',i[1])
```

OUTPUT

Create

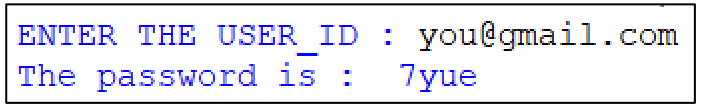


User_Id	Password
we@gmail.com	67ty
you@gmail.com	7yue
get@gmail.com	tyh2
yoga@gmail.com	2e3r
me@gmail.com	2yg8



```
How many enteries are there : 5
Input no. : 1
ENTER THE USER ID : we@gmail.com
ENTER THE PASSWORD : 67ty
Input no. : 2
ENTER THE USER ID : you@gmail.com
ENTER THE PASSWORD : 7yue
Input no. : 3
ENTER THE USER ID : get@gmail.com
ENTER THE PASSWORD : tyh2
Input no. : 4
ENTER THE USER ID : yoga@gmail.com
ENTER THE PASSWORD : 2e3r
Input no. : 5
ENTER THE USER ID : me@gmail.com
ENTER THE PASSWORD : 2yg8
>>>
```

Search



```
ENTER THE USER_ID : you@gmail.com
The password is :  7yue
```

Riya Verma , XII B

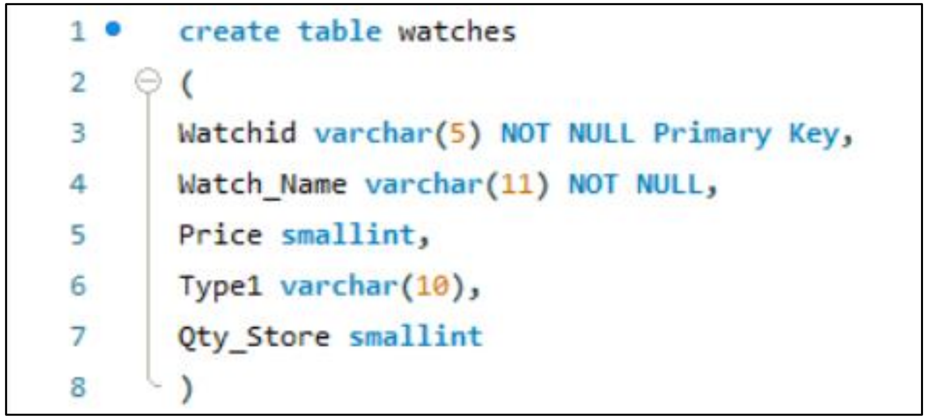
MySQL Queries

Sno.	Table Name	My SQL Queries
Query1.	Watches	a) Create a table named watches b) Insert 5 tuples into the table c) Display watch’s name and price of those watches which have price range in between 5000-15000 d) Display average quantity of each type of watch in the store e) Update the Watchid of ‘Golden Time’ to ‘W009’ f) Remove tuples whose names begin with ‘H’
Query 2.	Teacher & Posting	a) Create a table named Teacher and Posting b) Insert the Tuples in both the tables c) Display the maximum salary of each department d) Display name, bonus for each teacher where bonus is 10% of salary e) Display all details of teacher in descending order of salary
Query 3.	FACULTY & COURSES	a) Create tables named FACULTY and COURSES b) Insert tuples in both the tables c) Increase the fees of all courses by 500 of ‘system design’ course d) Display details of those courses which are taught by ‘Sulekha’ in decreasing order of courses e) Remove the tuples of those faculties whose salary is greater than 12000

Query 1.(a)

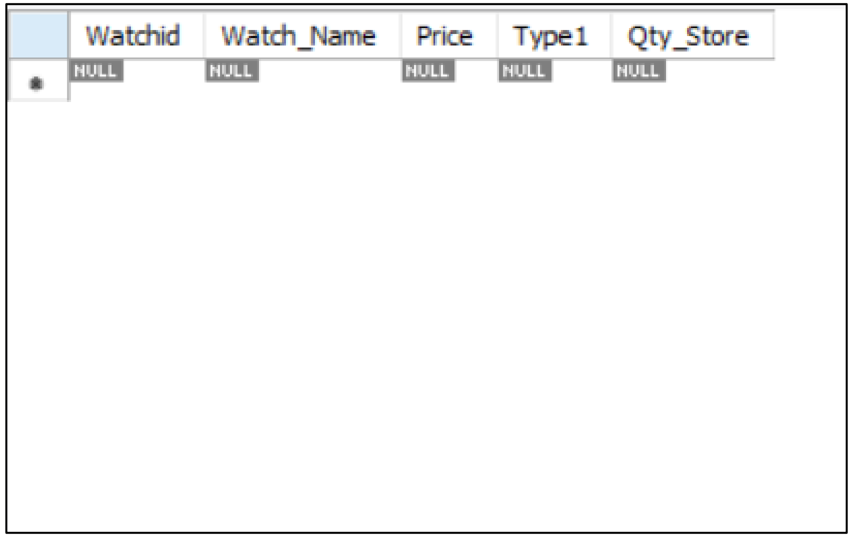
Create a table named watches

INPUT



```
create table watches
(
Watchid varchar(5) NOT NULL Primary Key,
Watch_Name varchar(11) NOT NULL,
Price smallint,
Type1 varchar(10),
Qty_Store smallint
)
```

OUTPUT

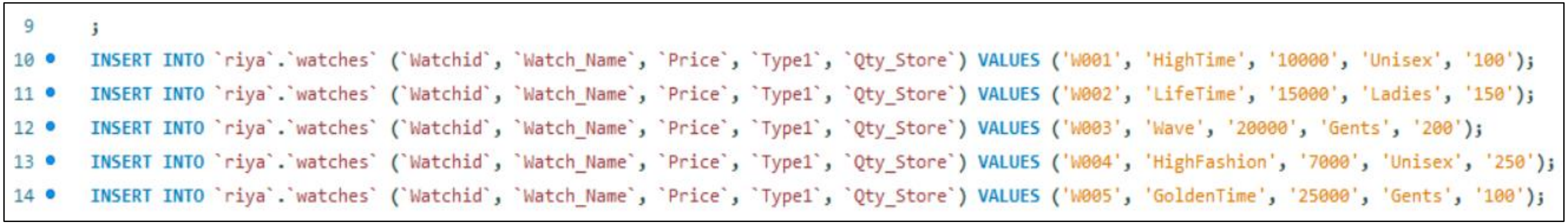


	Watchid	Watch_Name	Price	Type1	Qty_Store
*	NULL	NULL	NULL	NULL	NULL

Query 1.(b)

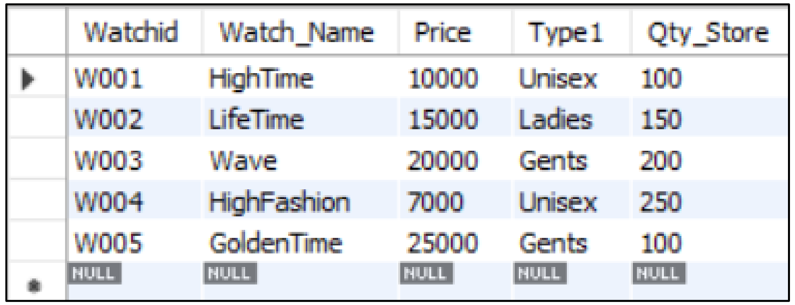
Insert 5 tuples into the table

INPUT



```
;
INSERT INTO `riya`.`watches` (`Watchid`, `Watch_Name`, `Price`, `Type1`, `Qty_Store`) VALUES ('W001', 'HighTime', '10000', 'Unisex', '100');
INSERT INTO `riya`.`watches` (`Watchid`, `Watch_Name`, `Price`, `Type1`, `Qty_Store`) VALUES ('W002', 'LifeTime', '15000', 'Ladies', '150');
INSERT INTO `riya`.`watches` (`Watchid`, `Watch_Name`, `Price`, `Type1`, `Qty_Store`) VALUES ('W003', 'Wave', '20000', 'Gents', '200');
INSERT INTO `riya`.`watches` (`Watchid`, `Watch_Name`, `Price`, `Type1`, `Qty_Store`) VALUES ('W004', 'HighFashion', '7000', 'Unisex', '250');
INSERT INTO `riya`.`watches` (`Watchid`, `Watch_Name`, `Price`, `Type1`, `Qty_Store`) VALUES ('W005', 'GoldenTime', '25000', 'Gents', '100');
```

OUTPUT

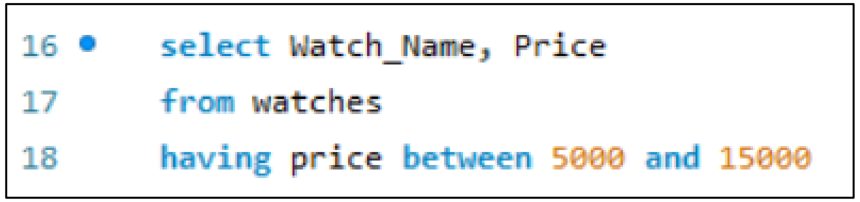


Watchid	Watch_Name	Price	Type1	Qty_Store
W001	HighTime	10000	Unisex	100
W002	LifeTime	15000	Ladies	150
W003	Wave	20000	Gents	200
W004	HighFashion	7000	Unisex	250
W005	GoldenTime	25000	Gents	100
NULL	NULL	NULL	NULL	NULL

Query 1.(c)

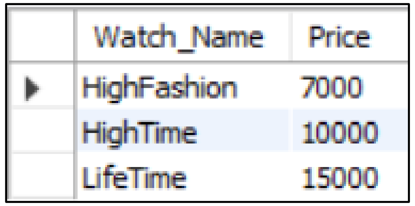
Display watch’s name and price of those watches which have price range in between 5000 – 15000

INPUT



```
select Watch_Name, Price
from watches
having price between 5000 and 15000
```

OUTPUT

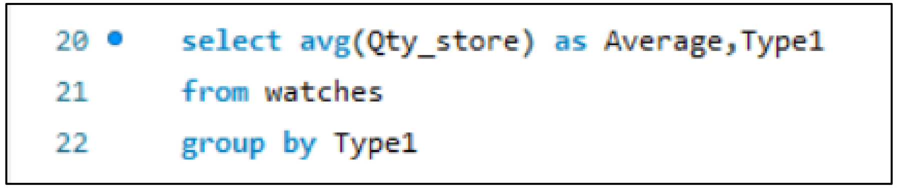


Watch_Name	Price
HighFashion	7000
HighTime	10000
LifeTime	15000

Query 1.(d)

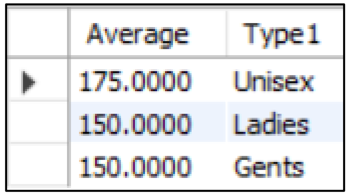
Display average quantity of each type of watch in the store

INPUT



```
select avg(Qty_store) as Average,Type1
from watches
group by Type1
```

OUTPUT

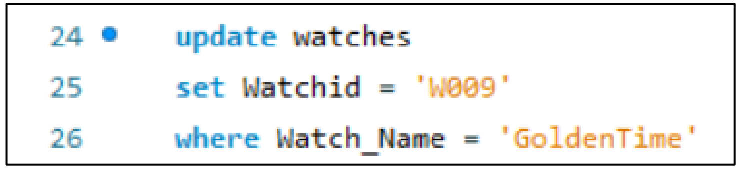


Average	Type1
175.0000	Unisex
150.0000	Ladies
150.0000	Gents

Query 1.(e)

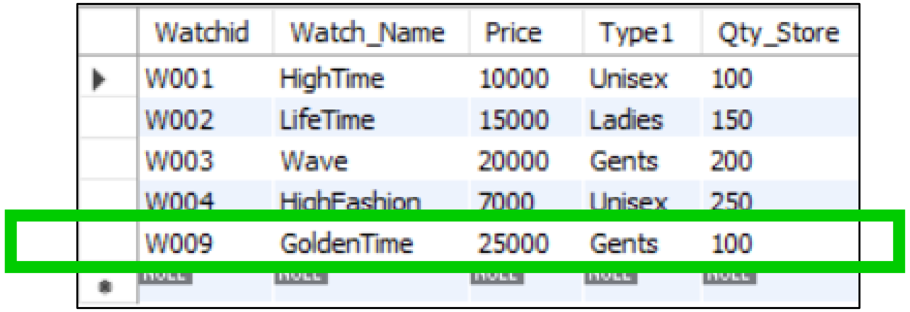
Update the Watchid of ‘Golden Time’ to ‘W009’

INPUT



```sql
update watches
set Watchid = 'W009'
where Watch_Name = 'GoldenTime'
```

OUTPUT

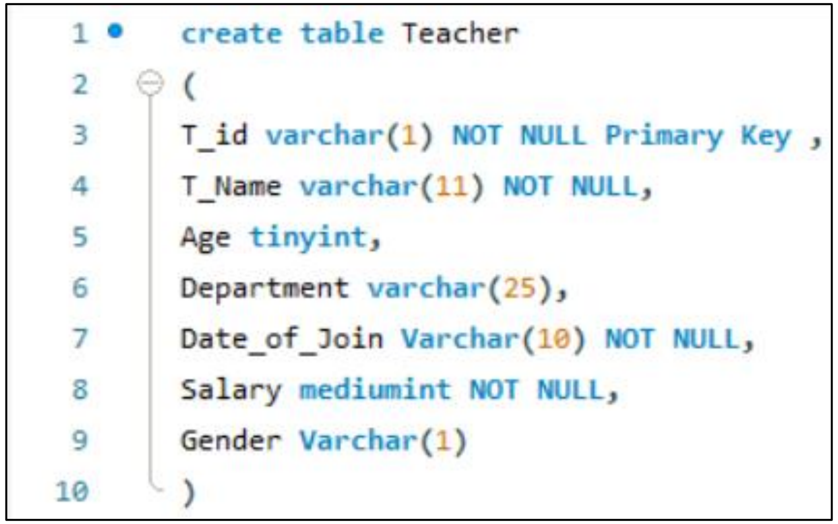


	Watchid	Watch_Name	Price	Type1	Qty_Store
▶	W001	HighTime	10000	Unisex	100
	W002	LifeTime	15000	Ladies	150
	W003	Wave	20000	Gents	200
	W004	HighFashion	7000	Unisex	250
	W009	GoldenTime	25000	Gents	100
*	NULL	NULL	NULL	NULL	NULL

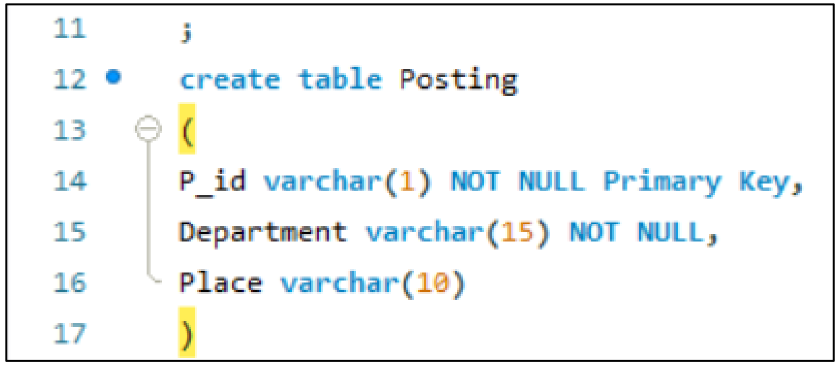
Query 2.(a)

Create a table named Teacher and Posting

INPUT



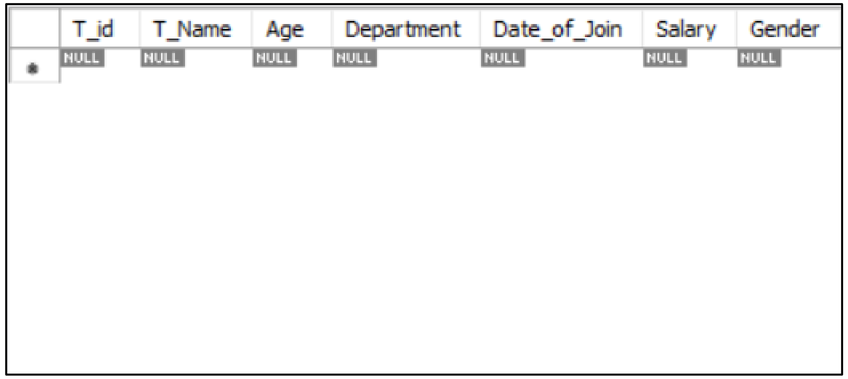
```
create table Teacher
(
T_id varchar(1) NOT NULL Primary Key ,
T_Name varchar(11) NOT NULL,
Age tinyint,
Department varchar(25),
Date_of_Join Varchar(10) NOT NULL,
Salary mediumint NOT NULL,
Gender Varchar(1)
)
```



```
;
create table Posting
(
P_id varchar(1) NOT NULL Primary Key,
Department varchar(15) NOT NULL,
Place varchar(10)
)
```

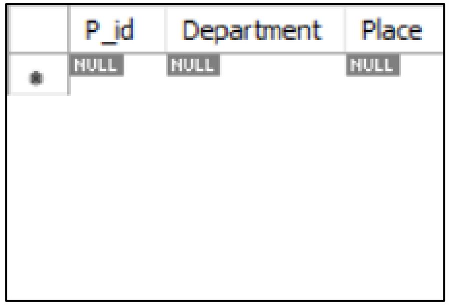
OUTPUT

Teacher



	T_id	T_Name	Age	Department	Date_of_Join	Salary	Gender
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Posting

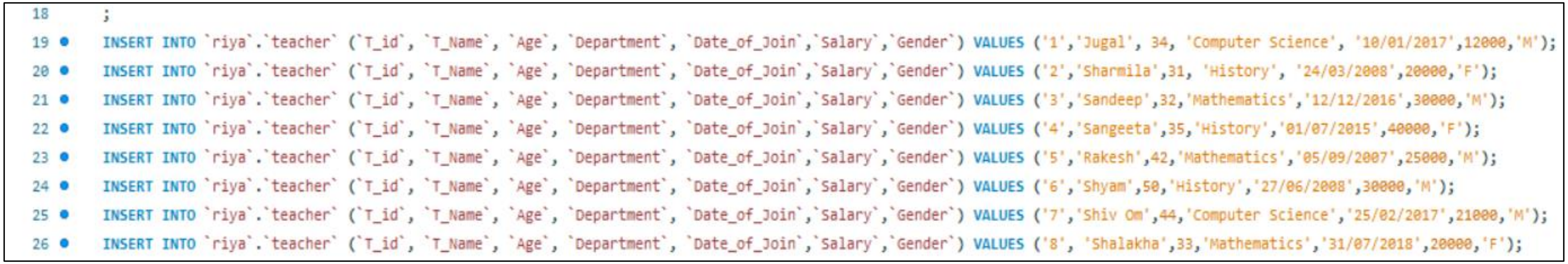


	P_id	Department	Place
*	NULL	NULL	NULL

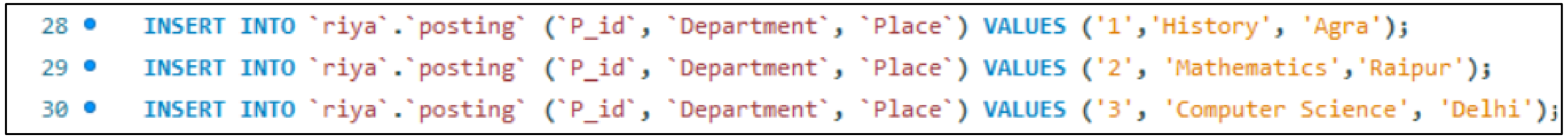
Query 2.(b)

Insert the Tuples in both the tables

INPUT



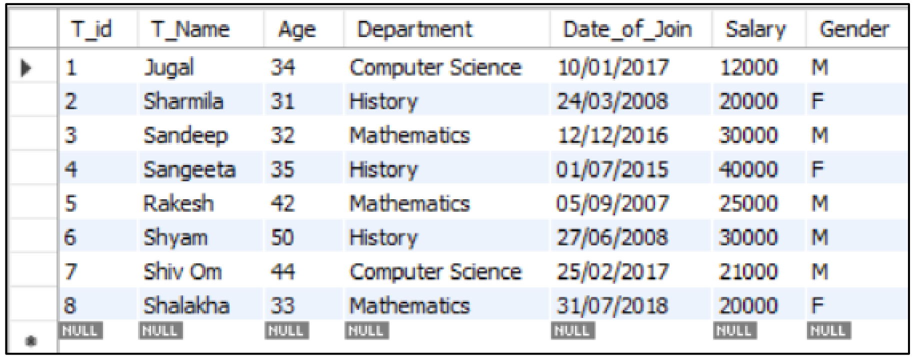
```
;
INSERT INTO `riya`.`teacher` (`T_id`, `T_Name`, `Age`, `Department`, `Date_of_Join`,`Salary`,`Gender`) VALUES ('1','Jugal', 34, 'Computer Science', '10/01/2017',12000,'M');
INSERT INTO `riya`.`teacher` (`T_id`, `T_Name`, `Age`, `Department`, `Date_of_Join`,`Salary`,`Gender`) VALUES ('2','Sharmila',31, 'History', '24/03/2008',20000,'F');
INSERT INTO `riya`.`teacher` (`T_id`, `T_Name`, `Age`, `Department`, `Date_of_Join`,`Salary`,`Gender`) VALUES ('3','Sandeep',32,'Mathematics','12/12/2016',30000,'M');
INSERT INTO `riya`.`teacher` (`T_id`, `T_Name`, `Age`, `Department`, `Date_of_Join`,`Salary`,`Gender`) VALUES ('4','Sangeeta',35,'History','01/07/2015',40000,'F');
INSERT INTO `riya`.`teacher` (`T_id`, `T_Name`, `Age`, `Department`, `Date_of_Join`,`Salary`,`Gender`) VALUES ('5','Rakesh',42,'Mathematics','05/09/2007',25000,'M');
INSERT INTO `riya`.`teacher` (`T_id`, `T_Name`, `Age`, `Department`, `Date_of_Join`,`Salary`,`Gender`) VALUES ('6','Shyam',50,'History','27/06/2008',30000,'M');
INSERT INTO `riya`.`teacher` (`T_id`, `T_Name`, `Age`, `Department`, `Date_of_Join`,`Salary`,`Gender`) VALUES ('7','Shiv Om',44,'Computer Science','25/02/2017',21000,'M');
INSERT INTO `riya`.`teacher` (`T_id`, `T_Name`, `Age`, `Department`, `Date_of_Join`,`Salary`,`Gender`) VALUES ('8', 'Shalakha',33,'Mathematics','31/07/2018',20000,'F');
```



```
INSERT INTO `riya`.`posting` (`P_id`, `Department`, `Place`) VALUES ('1','History', 'Agra');
INSERT INTO `riya`.`posting` (`P_id`, `Department`, `Place`) VALUES ('2', 'Mathematics','Raipur');
INSERT INTO `riya`.`posting` (`P_id`, `Department`, `Place`) VALUES ('3', 'Computer Science', 'Delhi');
```

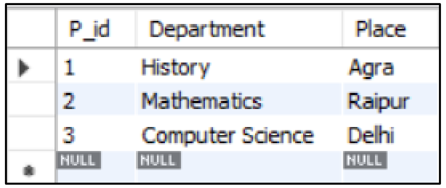
OUTPUT

Teacher



	T_id	T_Name	Age	Department	Date_of_Join	Salary	Gender
▶	1	Jugal	34	Computer Science	10/01/2017	12000	M
	2	Sharmila	31	History	24/03/2008	20000	F
	3	Sandeep	32	Mathematics	12/12/2016	30000	M
	4	Sangeeta	35	History	01/07/2015	40000	F
	5	Rakesh	42	Mathematics	05/09/2007	25000	M
	6	Shyam	50	History	27/06/2008	30000	M
	7	Shiv Om	44	Computer Science	25/02/2017	21000	M
	8	Shalakha	33	Mathematics	31/07/2018	20000	F
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Posting

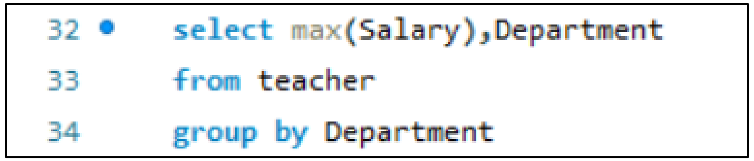


	P_id	Department	Place
▶	1	History	Agra
	2	Mathematics	Raipur
	3	Computer Science	Delhi
*	NULL	NULL	NULL

Query 2.(c)

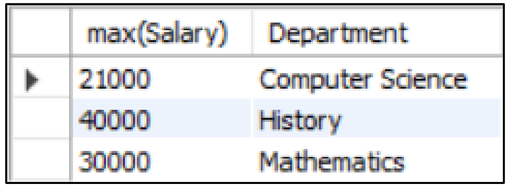
Display the maximum salary of each department

INPUT



```sql
select max(Salary),Department
from teacher
group by Department
```

OUTPUT

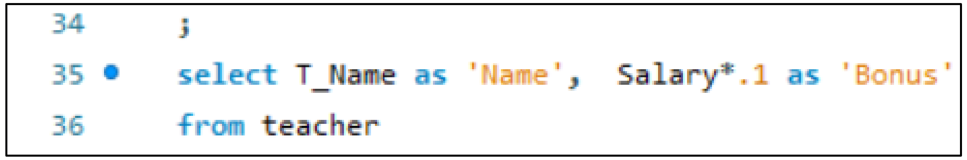


max(Salary)	Department
21000	Computer Science
40000	History
30000	Mathematics

Query 2.(d)

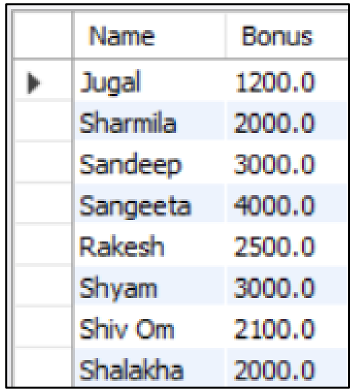
Display name, bonus for each teacher where bonus is 10% of salary

INPUT



```sql
;
select T_Name as 'Name',  Salary*.1 as 'Bonus'
from teacher
```

OUTPUT

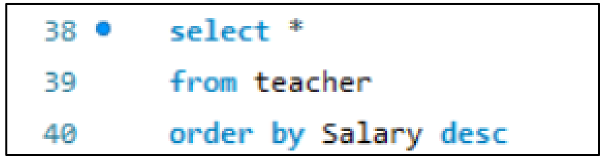


Name	Bonus
Jugal	1200.0
Sharmila	2000.0
Sandeep	3000.0
Sangeeta	4000.0
Rakesh	2500.0
Shyam	3000.0
Shiv Om	2100.0
Shalakha	2000.0

Query 2.(e)

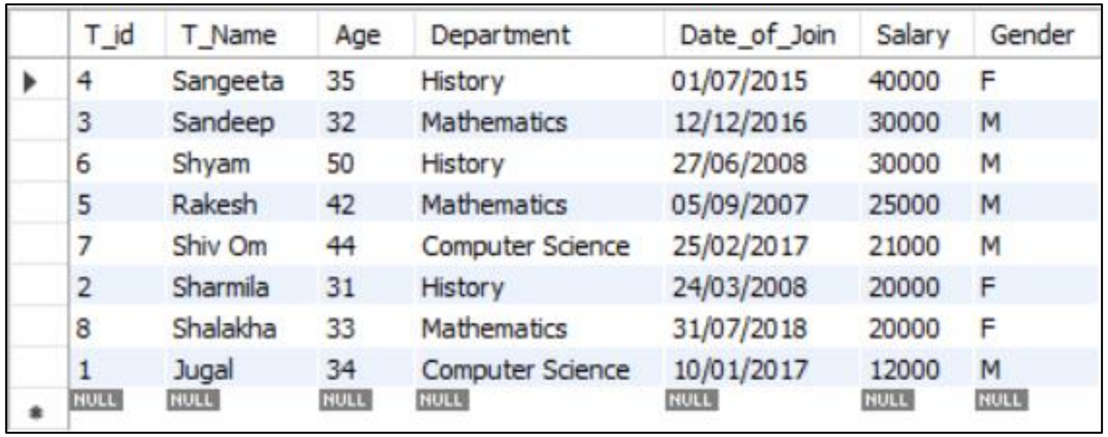
Display all details of teacher in descending order of salary

INPUT



```
select *
from teacher
order by Salary desc
```

OUTPUT

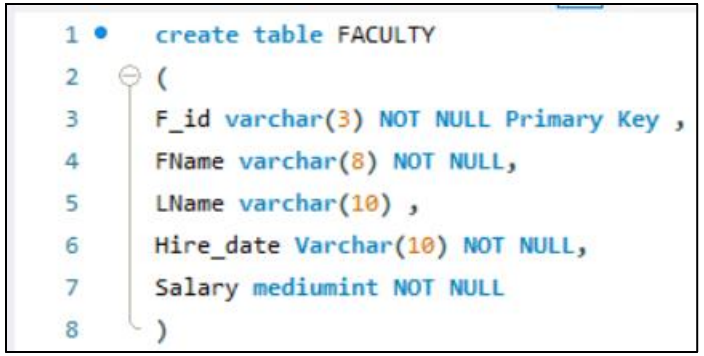


T_id	T_Name	Age	Department	Date_of_Join	Salary	Gender
4	Sangeeta	35	History	01/07/2015	40000	F
3	Sandeep	32	Mathematics	12/12/2016	30000	M
6	Shyam	50	History	27/06/2008	30000	M
5	Rakesh	42	Mathematics	05/09/2007	25000	M
7	Shiv Om	44	Computer Science	25/02/2017	21000	M
2	Sharmila	31	History	24/03/2008	20000	F
8	Shalakha	33	Mathematics	31/07/2018	20000	F
1	Jugal	34	Computer Science	10/01/2017	12000	M
NULL	NULL	NULL	NULL	NULL	NULL	NULL

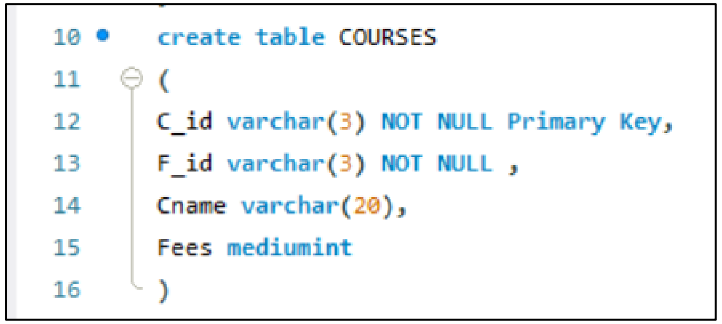
Query 3.(a)

Create tables named FACULTY and COURSES

INPUT



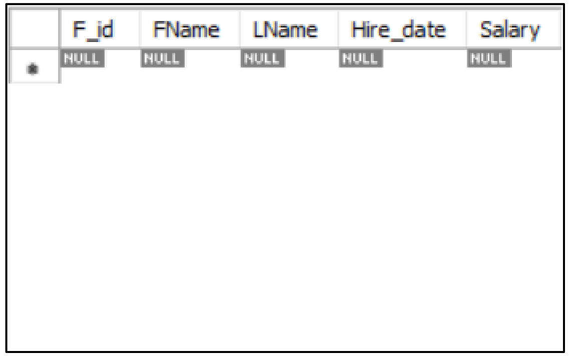
```
create table FACULTY
(
F_id varchar(3) NOT NULL Primary Key ,
FName varchar(8) NOT NULL,
LName varchar(10) ,
Hire_date Varchar(10) NOT NULL,
Salary mediumint NOT NULL
)
```



```
create table COURSES
(
C_id varchar(3) NOT NULL Primary Key,
F_id varchar(3) NOT NULL ,
Cname varchar(20),
Fees mediumint
)
```

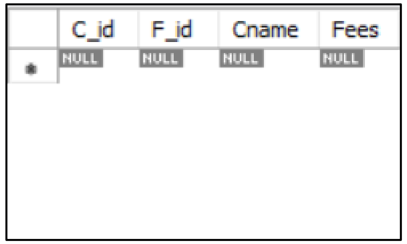
OUTPUT

FACULTY



	F_id	FName	LName	Hire_date	Salary
*	NULL	NULL	NULL	NULL	NULL

COURSES



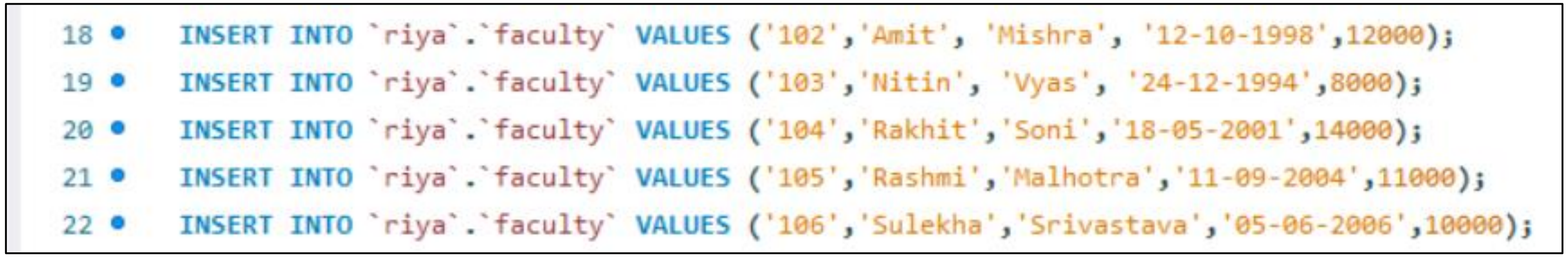
	C_id	F_id	Cname	Fees
*	NULL	NULL	NULL	NULL

Query 3. FACULTY & COURSES

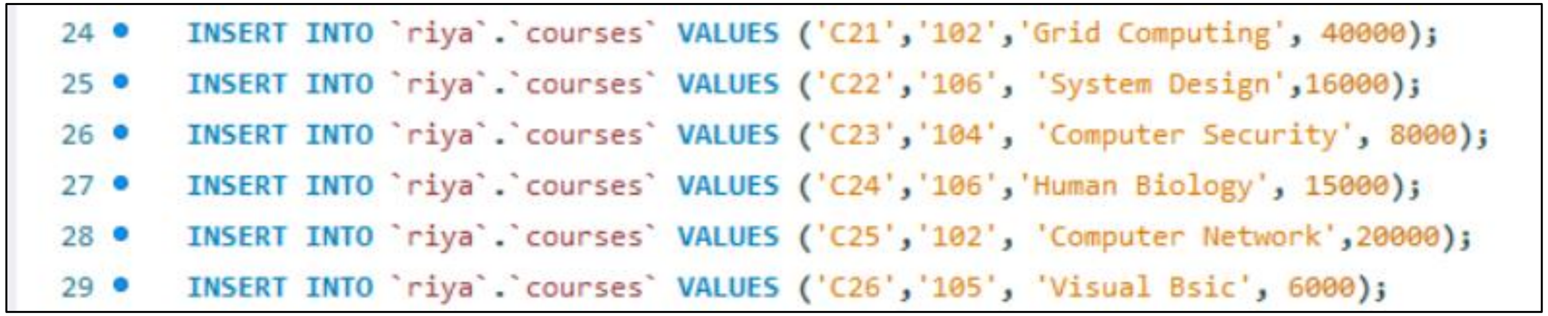
Query 3.(b)

Insert the Tuples in both the tables

INPUT



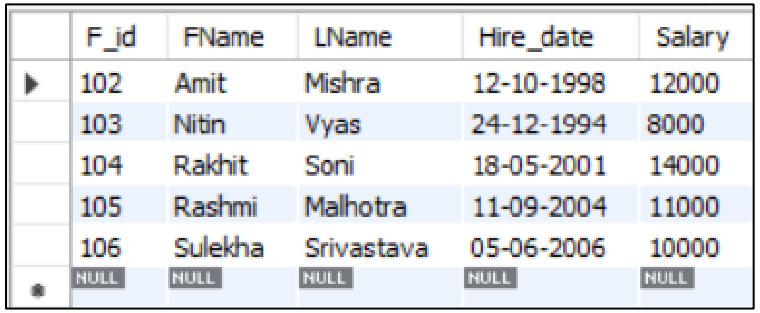
```sql
INSERT INTO `riya`.`faculty` VALUES ('102','Amit', 'Mishra', '12-10-1998',12000);
INSERT INTO `riya`.`faculty` VALUES ('103','Nitin', 'Vyas', '24-12-1994',8000);
INSERT INTO `riya`.`faculty` VALUES ('104','Rakhit','Soni','18-05-2001',14000);
INSERT INTO `riya`.`faculty` VALUES ('105','Rashmi','Malhotra','11-09-2004',11000);
INSERT INTO `riya`.`faculty` VALUES ('106','Sulekha','Srivastava','05-06-2006',10000);
```



```sql
INSERT INTO `riya`.`courses` VALUES ('C21','102','Grid Computing', 40000);
INSERT INTO `riya`.`courses` VALUES ('C22','106', 'System Design',16000);
INSERT INTO `riya`.`courses` VALUES ('C23','104', 'Computer Security', 8000);
INSERT INTO `riya`.`courses` VALUES ('C24','106','Human Biology', 15000);
INSERT INTO `riya`.`courses` VALUES ('C25','102', 'Computer Network',20000);
INSERT INTO `riya`.`courses` VALUES ('C26','105', 'Visual Bsic', 6000);
```

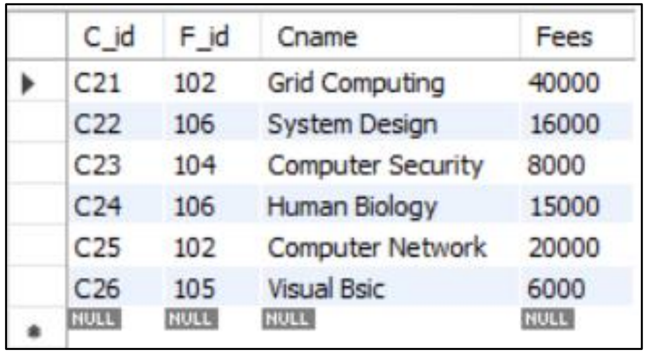
OUTPUT

FACULTY



F_id	FName	LName	Hire_date	Salary
102	Amit	Mishra	12-10-1998	12000
103	Nitin	Vyas	24-12-1994	8000
104	Rakhit	Soni	18-05-2001	14000
105	Rashmi	Malhotra	11-09-2004	11000
106	Sulekha	Srivastava	05-06-2006	10000
NULL	NULL	NULL	NULL	NULL

COURSES



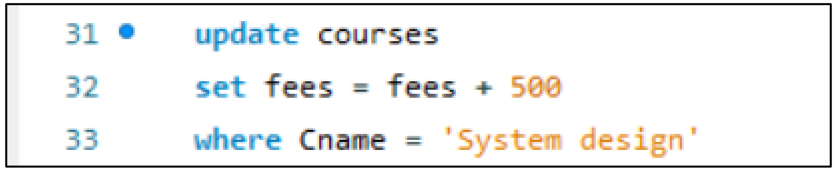
C_id	F_id	Cname	Fees
C21	102	Grid Computing	40000
C22	106	System Design	16000
C23	104	Computer Security	8000
C24	106	Human Biology	15000
C25	102	Computer Network	20000
C26	105	Visual Bsic	6000
NULL	NULL	NULL	NULL

Riya Verma , XII B

Query 3.(c)

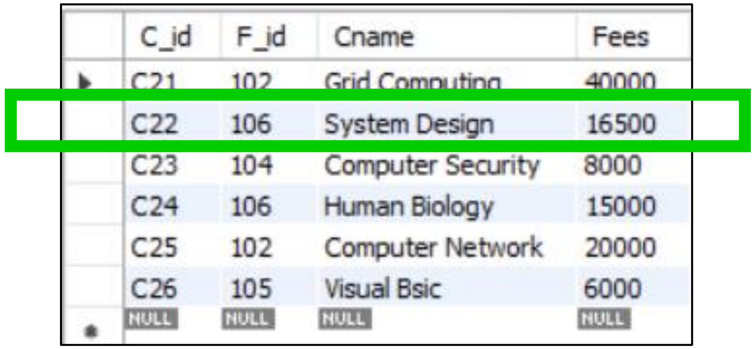
Increase the fees of all courses by 500 of ‘system design’ course

INPUT



```sql
update courses
set fees = fees + 500
where Cname = 'System design'
```

OUTPUT

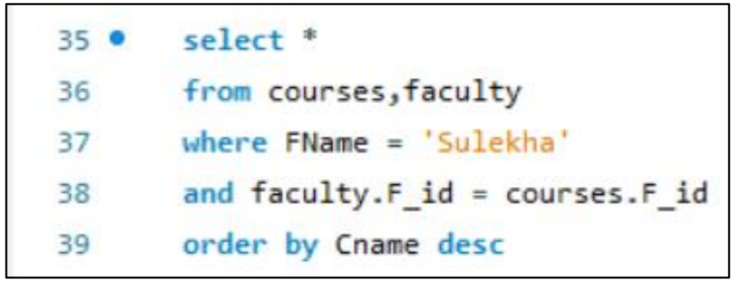


	C_id	F_id	Cname	Fees
▶	C21	102	Grid Computing	40000
	C22	106	System Design	16500
	C23	104	Computer Security	8000
	C24	106	Human Biology	15000
	C25	102	Computer Network	20000
	C26	105	Visual Bsic	6000
*	NULL	NULL	NULL	NULL

Query 3.(d)

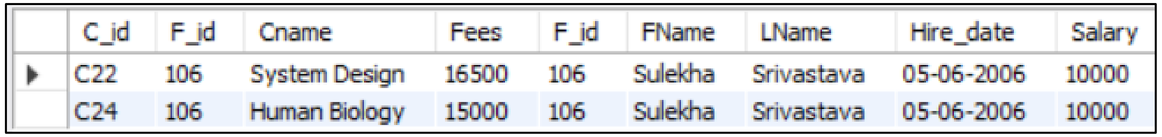
Display details of those courses which are taught by ‘Sulekha’ in decreasing order of courses

INPUT



```sql
select *
from courses,faculty
where FName = 'Sulekha'
and faculty.F_id = courses.F_id
order by Cname desc
```

OUTPUT

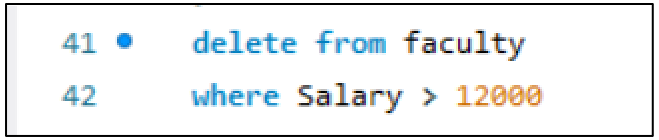


	C_id	F_id	Cname	Fees	F_id	FName	LName	Hire_date	Salary
▶	C22	106	System Design	16500	106	Sulekha	Srivastava	05-06-2006	10000
	C24	106	Human Biology	15000	106	Sulekha	Srivastava	05-06-2006	10000

Query 3.(e)

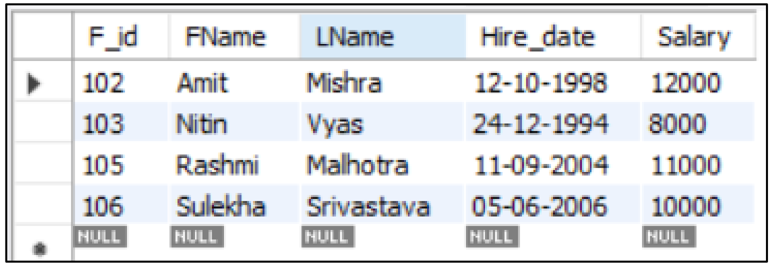
Remove the tuples of those faculties whose salary is greater than 12000

INPUT



```
delete from faculty
where Salary > 12000
```

OUTPUT



	F_id	FName	LName	Hire_date	Salary
▶	102	Amit	Mishra	12-10-1998	12000
	103	Nitin	Vyas	24-12-1994	8000
	105	Rashmi	Malhotra	11-09-2004	11000
	106	Sulekha	Srivastava	05-06-2006	10000
*	NULL	NULL	NULL	NULL	NULL