Class XII

Computer Science (083)

Marking Scheme

Time Allowed: 3 hours **MM: 70**

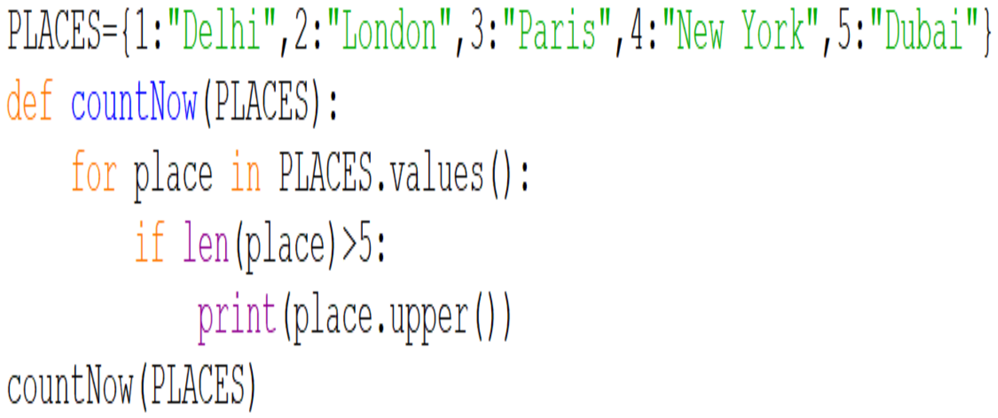
Ques No	Question and Answers	Distribution of Marks	Total Marks
SECTION A			
1	False	1 mark for correct answer	1
2	Option b 6,20	1 mark for correct answer	1
3	Option c -244.0	1 mark for correct answer	1
4	PYTHON-is-Fun	1 mark for correct answer	1
5	Option b 8,15	1 mark for correct answer	1
6	Option a PAN	1 mark for correct answer	1
7	Option b `del D1["Red"]`	1 mark for correct answer	1
8	Option b	1 mark for correct answer	1

	ceieP0		
9	Option d Statement 4	1 mark for correct answer	1
10	Option b WHITE* BLACK*	1 mark for correct answer	1
11	Option b Modulator	1 mark for correct answer	1
12	Option c global b	1 mark for correct answer	1
13	True	1 mark for correct answer	1
14	Option c A candidate key that is not a primary key is a foreign key.	1 mark for correct answer	1
15	Circuit	1 mark for correct answer	1
16	Option c `seek()`	1 mark for correct answer	1
17	Option d A is false but R is True	1 mark for correct answer	1

18	Option b Both A and R are true but R is not the correct explanation for A	1 mark for correct answer	1
	SECTION B		
19	(i) POP3 – Post Office Protocol 3 URL – Uniform Resource Locator (ii) HTML(Hyper text mark Up language) • We use pre-defined tags • Static web development language – only focuses on how data looks • It use for only displaying data, cannot transport data • Not case sensistive XML (Extensible Markup Language) • we can define our own tags and use them • Dynamic web development language – as it is used for transporting and storing data • Case sensitive OR (i) Bandwidth is the maximum rate of data transfer over a given transmission medium. / The amount of information that can be transmitted over a network.	½ mark for each correct expansion 1 mark for any one correct difference No mark to be awarded if only full form is given 1 mark for correct definition	1+1=2

	(ii) https (Hyper Text Transfer Protocol Secure) is the protocol that uses SSL (Secure Socket Layer) to encrypt data being transmitted over the Internet. Therefore, https helps in secure browsing while http does not.	1 mark for correct difference.	

20	**def** revNumber(num):
	    rev = 0
	    rem = 0
	    while num > 0:
	        rem **=n**um %10
	        rev = rev*10 + rem
	        num = num//10
	    return rev
	print(revNumber(1234))
		½ mark for each correction made	2

21	
	PLACES={1:"Delhi",2:"London",3:"Paris",4:"New York",5:"Dubai"}
	def countNow(PLACES):
	    for place in PLACES.values():
	        if len(place)>5:
	            print(place.upper())
	countNow(PLACES)
	
	OR
		½ mark for correct function header
		½ mark for correct loop
		½ mark for correct if statement
		½ mark for displaying the output
		½ mark for correct function header
		½ mark for using split()
			2

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

```
def lenWords(STRING):
    T=()
    L=STRING.split()
    for word in L:
        length=len(word)
        T=T+(length,)
    return T
```

Note: Any other correct logic may be marked
</td><td>½ mark for adding to tuple
½ mark for return statement</td><td></td></tr>
<tr><td>22</td><td>4*L
33*4
21*S
10*6</td><td>½ mark for each correct line of output</td><td>2</td></tr>
<tr><td>23</td><td>

```
  (i)  L1.insert(2,200)

  (ii) message.endswith('.')
```

OR

```
import statistics
print(statistics.mode(studentAge) )
```
</td><td>1 mark for each correct statement
1 mark for correct import statement
1 mark for correct command with mode() and print()</td><td>1+1=2</td></tr>
<tr><td>24</td><td>SQL Command to add primary key:

```
ALTER TABLE Employee ADD EmpId INTEGER
PRIMARY KEY;
```
</td><td>1 mark for correct ALTER TABLE command</td><td>2</td></tr>
</table>

	As the primary key is added as the last field, the command for inserting data will be:
	`INSERT INTO Employee`
	`VALUES("Shweta","Production",26900,999);`
	Alternative answer:
	`INSERT INTO`
	`Employee(EmpId,Ename,Department,Salary)`
	`VALUES(999,"Shweta","Production",26900);`
	OR
	To delete the attribute, category:
	`ALTER TABLE Sports`
	`DROP category;`
	To add the attribute, TypeSport
	`ALTER TABLE Sports`
	`ADD TypeSport char(10) NOT NULL;`
		1 mark for correct INSERT command
		1 mark for correct ALTER TABLE command with DROP
		1 mark for correct ALTER TABLE command with ADD	

25	10.0$20 10.0$2.0###	1 mark for each correct line of output	2
SECTION C			
26	ND-*34	½ mark for each correct character	3
27			
	(i) **COUNT(DISTINCT SPORTS)**	1 mark for each correct output	1*3=3

4

(ii)

CNAME	SPORTS
AMINA	CHESS

(iii)

CNAME	AGE	PAY
AMRIT	28	1000
VIRAT	35	1050

28

```
def test():
    fObj1 = open("Alpha.txt","r")
    data = fObj1.readlines()
    for line in data:
        L=line.split()
        if L[0]=="You":
            print(line)
    fObj1.close()
```

1 mark for correctly opening and closing files

½ mark for correctly reading data

1 mark for correct loop and if statement

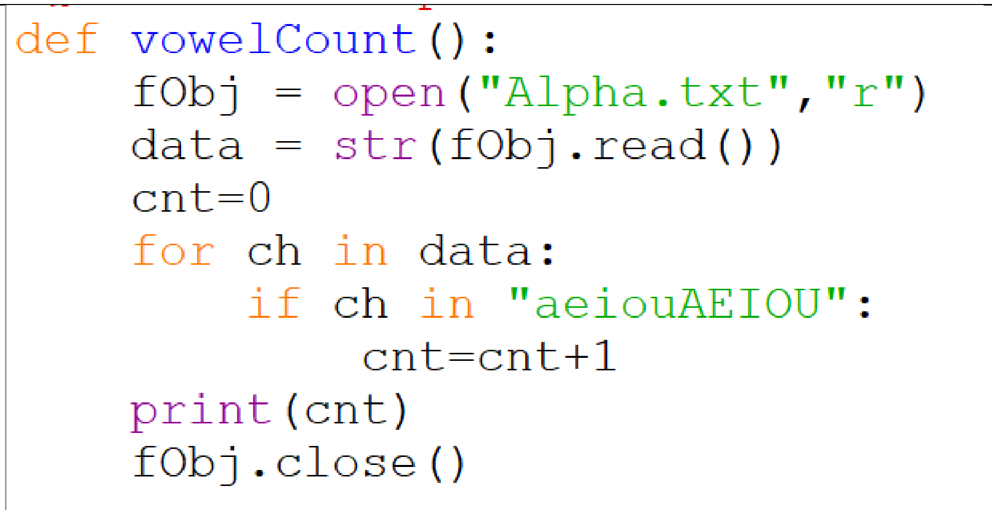
½ mark for displaying data

3

OR

1 mark for correctly opening and closing the files

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



```
def vowelCount():
    fObj = open("Alpha.txt","r")
    data = str(fObj.read())
    cnt=0
    for ch in data:
        if ch in "aeiouAEIOU":
            cnt=cnt+1
    print(cnt)
    fObj.close()
```

<u>Note: Any other correct logic may be marked</u>

</td>
<td>½ mark for correctly reading data
1 mark for correct loop and if statement
½ mark for displaying the output.</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>

(i)

```
UPDATE Personal
SET Salary=Salary + Salary*0.5
WHERE Allowance IS NOT NULL;
```

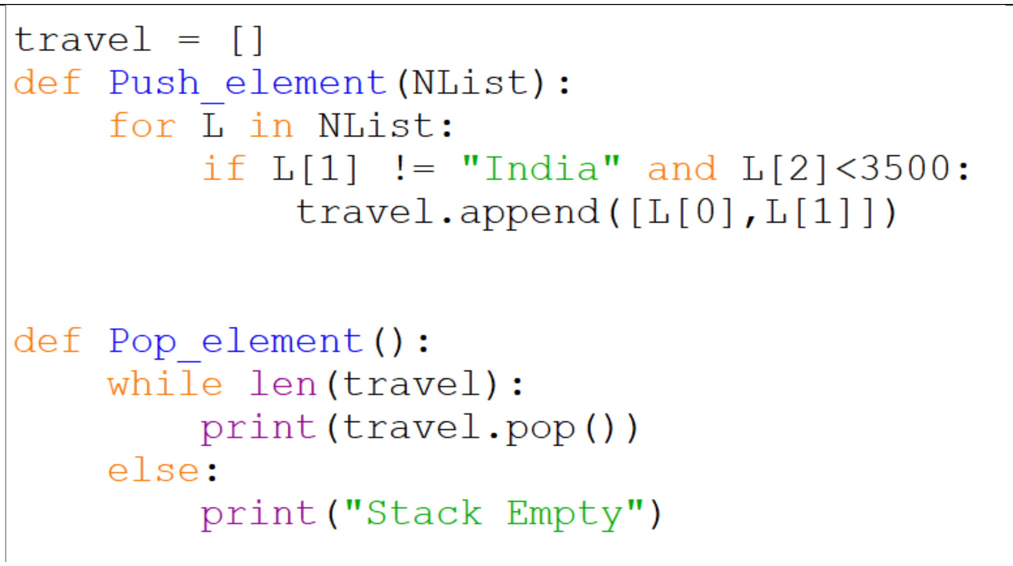
(ii)

```
SELECT Name, Salary + Allowance AS
"Total Salary" FROM Personal;
```

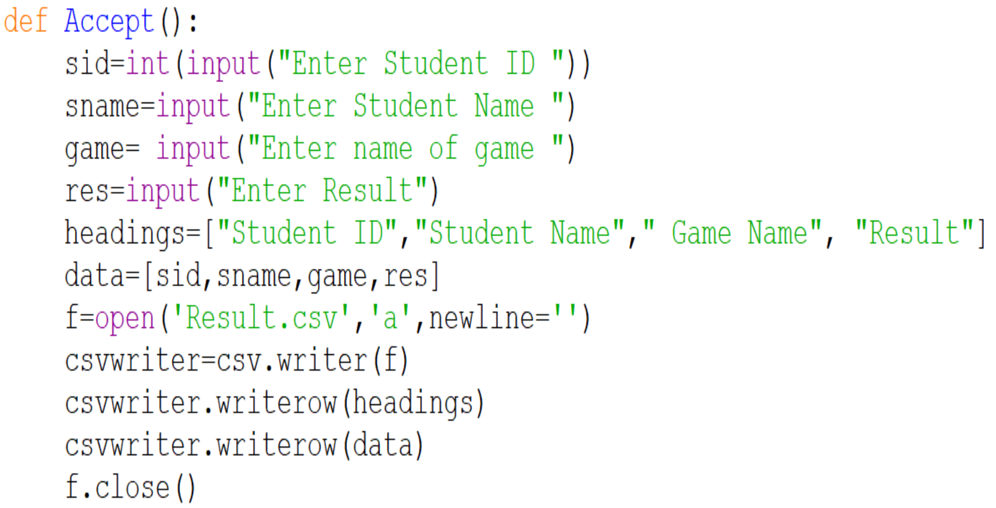
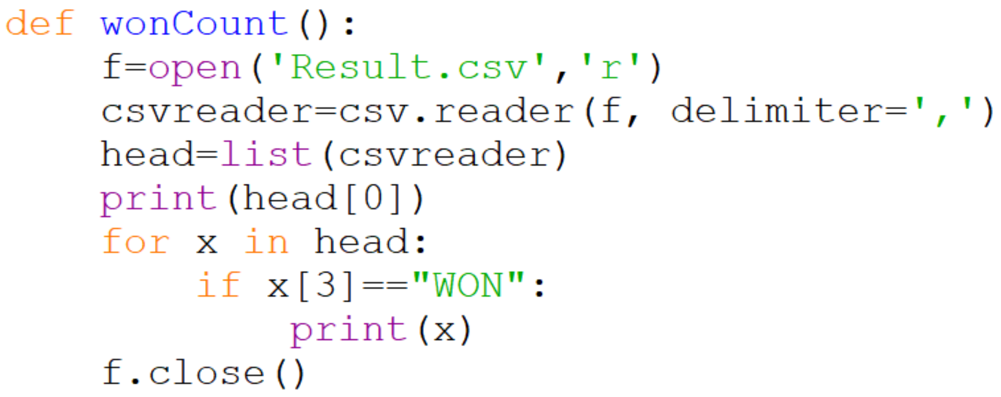
(iii)

```
DELETE FROM Personal
WHERE Salary>25000
```

</td>
<td>1 mark for each correct query</td>
<td>1*3=3</td>
</tr>
</table>

<table>
<tr>
<td>30</td>
<td>
<pre>
travel = []
def Push_element(NList):
    for L in NList:
        if L[1] != "India" and L[2]<3500:
            travel.append([L[0],L[1]])

def Pop_element():
    while len(travel):
        print(travel.pop())
    else:
        print("Stack Empty")
</pre></td>
<td>1 ½ marks for each function</td>
<td>3</td>
</tr>
<tr>
<td colspan="4"><b><u>SECTION D</u></b></td>
</tr>
<tr>
<td>31</td>
<td><pre>
  (i)
      SELECT PName, BName FROM PRODUCT P,
      BRAND B WHERE P.BID=B.BID;
  (ii)
      DESC PRODUCT;
  (iii)
      SELECT BName, AVG(Rating) FROM PRODUCT
      P, BRAND B
      WHERE P.BID=B.BID
      GROUP BY BName
      HAVING BName='Medimix' OR
      BName='Dove';
  (iv)
      SELECT PName, UPrice, Rating
      FROM PRODUCT
      ORDER BY Rating DESC;
</pre></td>
<td>1 mark for each correct query</td>
<td>1*4=4</td>
</tr>
</table>

<table>
<tr><td>32</td><td>

<pre>
def Accept():
    sid=int(input("Enter Student ID "))
    sname=input("Enter Student Name ")
    game= input("Enter name of game ")
    res=input("Enter Result")
    headings=["Student ID","Student Name"," Game Name", "Result"]
    data=[sid,sname,game,res]
    f=open('Result.csv','a',newline='')
    csvwriter=csv.writer(f)
    csvwriter.writerow(headings)
    csvwriter.writerow(data)
    f.close()
</pre>

<pre>
def wonCount():
    f=open('Result.csv','r')
    csvreader=csv.reader(f, delimiter=',')
    head=list(csvreader)
    print(head[0])
    for x in head:
        if x[3]=="WON":
            print(x)
    f.close()
</pre>
</td><td>½ mark for accepting data correctly

½ mark for opening and closing file

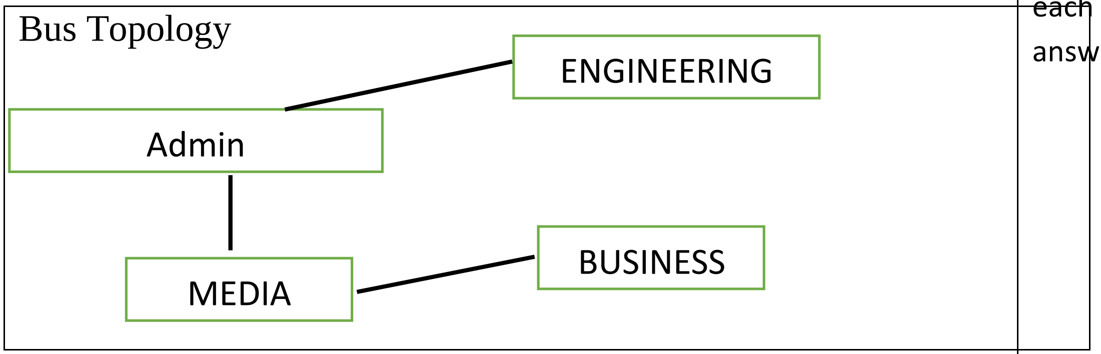
½ mark for writing headings

½ mark for writing row

½ mark for opening and closing file

½ mark for reader object

½ mark for print heading

½ mark for printing data</td><td>4</td></tr>
<tr><td colspan="4"><b><u>SECTION E</u></b></td></tr>
<tr><td>33</td><td>a)


</td><td>1 mark for each correct answer</td><td>1*5=5</td></tr>
</table>

	b) Switch c) Admin block, as it has maximum number of computers. d) Microwave e) No, a repeater is not required in the given cable layout as the length of transmission medium between any two blocks does not exceed 70 m.		
34	(i) r+ mode: • Primary function is reading • File pointer is at beginning of file • if the file does not exist, it results in an error w+ mode: • primary function is writing • if the file does not exist, it creates a new file. • If the file exists, previous data is overwritten • File pointer is at the beginning of file (ii)	1 mark for each correct difference (minimum two differences should be given) ½ mark for correctly opening and closing files ½ mark for correct try and except block ½ mark for correct loop 1 mark for correctly copying data	2+3=5

```
def copyData():
    fObj = open("SPORT.DAT", "rb")
    fObj1 = open("BASKET.DAT","wb")
    cnt=0
    try:
        while True:
            data = pickle.load(fObj)
            print(data)
            if data[0] == "Basket Ball":
                pickle.dump(data,fObj1)
                cnt+=1
    except:
        fObj.close()
        fObj1.close()
    return cnt
```

½ mark for correct return statement

½ mark for correctly opening and closing files

½ mark for correct try and except block

½ mark for correct loop

½ mark for correct if statement

1 mark for correctly displaying data

OR

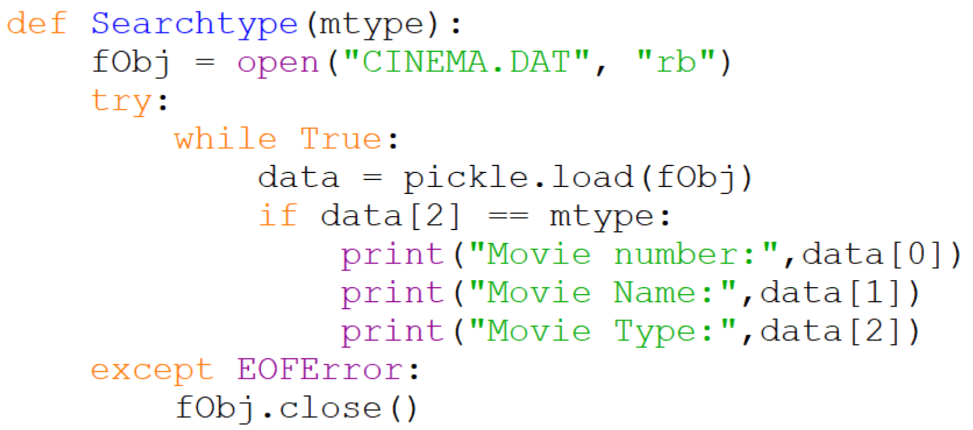
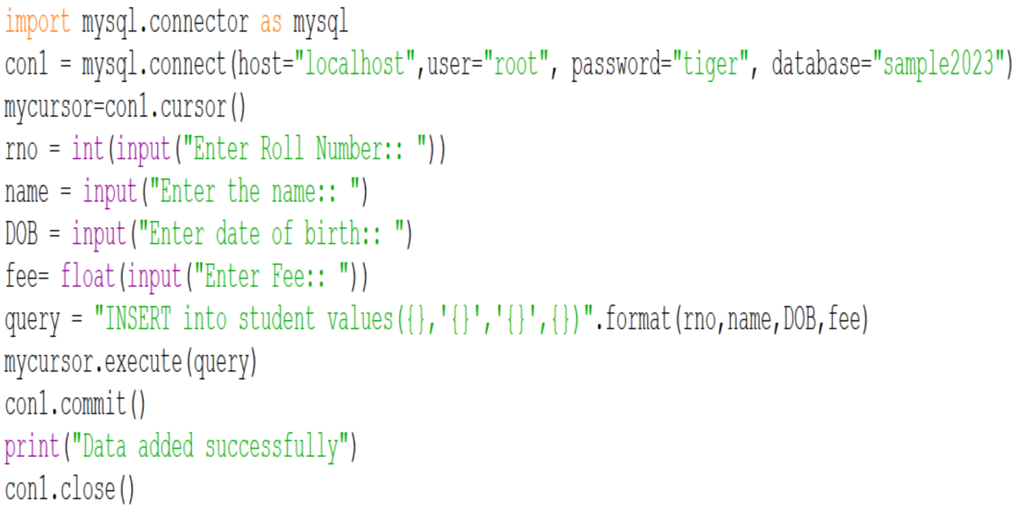
(i) Text files:

- Extension is .txt
- Data is stored in ASCII format that is human readable
- Has EOL character that terminates each line of data stored in the text files

Binary Files

- Extension is .dat
- Data is stored in binary form (0s and 1s), that is not human readable.

(ii)

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

<pre>
def Searchtype(mtype):
    fObj = open("CINEMA.DAT", "rb")
    try:
        while True:
            data = pickle.load(fObj)
            if data[2] == mtype:
                print("Movie number:",data[0])
                print("Movie Name:",data[1])
                print("Movie Type:",data[2])
    except EOFError:
        fObj.close()
</pre>
<b>Note: Any other correct logic may be marked</b>
</td>
<td></td>
<td></td>
</tr>
<tr>
<td>35</td>
<td>
(i) Domain is a set of values from which an attribute can take value in each row. For example, roll no field can have only integer values and so its domain is a set of integer values
(ii)

<pre>
import mysql.connector as mysql
con1 = mysql.connect(host="localhost",user="root", password="tiger", database="sample2023")
mycursor=con1.cursor()
rno = int(input("Enter Roll Number:: "))
name = input("Enter the name:: ")
DOB = input("Enter date of birth:: ")
fee= float(input("Enter Fee:: "))
query = "INSERT into student values({},'{}','{}',{})".format(rno,name,DOB,fee)
mycursor.execute(query)
con1.commit()
print("Data added successfully")
con1.close()
</pre>
<b>Note: Any other correct logic may be marked</b>
</td>
<td>
½ mark for correct definition
½ mark for correct example
½ mark for importing correct module
1 mark for correct connect()
½ mark for correctly accepting the input
1 ½ mark for correctly
</td>
<td>1+4=5</td>
</tr>
</table>

executing the query

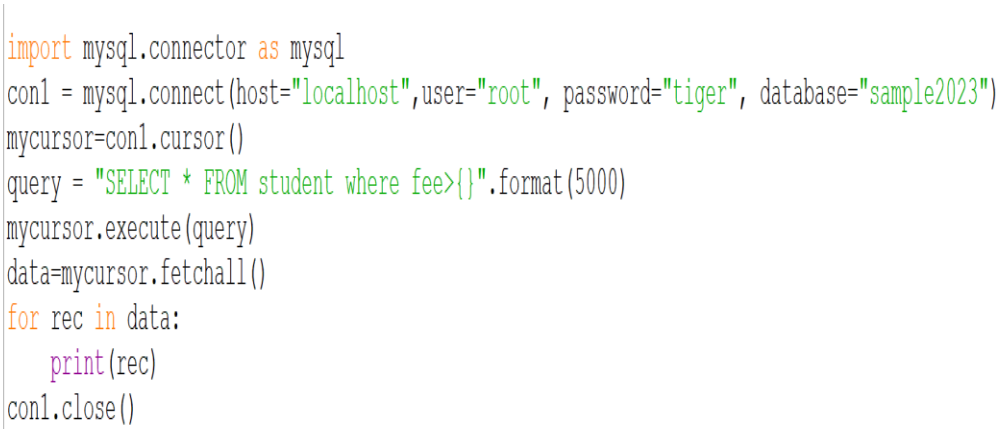
½ mark for correctly using commit()

OR

(i) All keys that have the properties to become a primary key are candidate keys. The candidate keys that do not become primary keys are alternate keys.

1 mark for correct difference

(ii)



```
import mysql.connector as mysql
con1 = mysql.connect(host="localhost",user="root", password="tiger", database="sample2023")
mycursor=con1.cursor()
query = "SELECT * FROM student where fee>{}".format(5000)
mycursor.execute(query)
data=mycursor.fetchall()
for rec in data:
    print(rec)
con1.close()
```

½ mark for importing correct module

1 mark for correct connect()

1 mark for correctly executing the query

½ mark for correctly using fetchall()

1 mark for correctly

		displaying data	