MARKING SCHEME

Class: XII Session: 2024-25

Computer Science (083)

Time allowed: 3 Hours **Maximum Marks: 70**

Q No.	SECTION A (21X1=21)	Marks
1.	False *(1 mark for correct answer)*	(1)
2.	(A) #THONPROGRAM *(1 mark for correct answer)*	(1)
3.	(A) not (True) and False *(1 mark for correct answer)*	(1)
4.	(B) ['I', 'ter', 'atio', 'al'] *(1 mark for correct answer)*	(1)
5.	ce lo *(1 mark for correct answer)*	(1)
6.	(B) False *(1 mark for correct answer)*	(1)
7.	(B) print(my_dict['apple', 'banana']) *(1 mark for correct answer)*	(1)
8.	(B) Removes the first occurrence of value x from the list *(1 mark for correct answer)*	(1)
9.	(C) 3 *(1 mark for correct answer)*	(1)
10.	file.seek(0) (OR file.seek(0,0)) *(1 mark for correct answer)*	(1)
11.	False *(1 mark for correct answer)*	(1)
12.	(C) 12#15% *(1 mark for correct answer)*	(1)
13.	Alter (or Alter Table) *(1 mark for correct answer)*	(1)

14.	(A) Details of all products whose names start with 'App' *(1 mark for correct answer)*	(1)
15.	(D) CHAR *(1 mark for correct answer)*	(1)
16.	(B) count() *(1 mark for correct answer)*	(1)
17.	(B) FTP *(1 mark for correct answer)*	(1)
18.	(B) Gateway *(1 mark for correct answer)*	(1)
19.	(B) Packet Switching *(1 mark for correct answer)*	(1)
20.	(C) A is True but R is False. *(1 mark for correct answer)*	(1)
21.	(C) A is True but R is False. *(1 mark for correct answer)*	(1)

Q No.	**SECTION B (7 X 2 =14)**	**Marks**
22.	A mutable object can be updated whereas an immutable object cannot be updated. Mutable object: [1,2] or {1:1,2:2} (Any one) Immutable object: (1,2) or '123' (Any one) *(1 mark for correct difference)* *(½ x 2 = 1 Mark for selecting correct objects)*	(2)
23.	(I) Arithmetic operators: +,- (II) Relational operators: >, >= *(½ x 4 = 2 Marks for each correct operator)*	(2)
24.	(I) A) L1.count(4) OR B) L1.sort() *(1 mark for correct answer)*	(2)

	(II) A) L1.extend(L2) OR B) L2.reverse() *(1 mark for correct answer)*	
25.	(A), (C) *(½ x 2 = 1 Mark)* Minimum and maximum possible values of the variable b: 1,6 *(½ x 2 = 1 Mark)*	(2)
26.	def swap_first_last(tup**):** if len(tup) < 2: **return tup** new_tup = (tup[-1],) + tup[1:-1] + (tup**[0],)** return new_tup result = swap_first_last((1, 2, 3, 4)) print("Swapped **tuple:", result)** (½ mark each for correcting 4 mistakes)	(2)
27.	(I) A) UNIQUE **OR** B) NOT NULL *(1 mark for correct answer)* (II) A) ALTER TABLE MOBILE DROP PRIMARY KEY; OR B) ALTER TABLE MOBILE ADD PRIMARY KEY (M_ID); *(1 mark for correct answer)*	(2)
28.	A) Advantage: Network extension is easy. Disadvantage: Failure of switch/hub results in failure of the network. *(1 mark for correct Advantage)* *(1 mark for correct Disadvantage)* **OR**	(2)

	B) SMTP: Simple Mail Transfer Protocol. SMTP is used for sending e-mails from client to server. *(1 mark for correct expansion)* *(1 mark for correct usage)*	

Q No.	SECTION C (3 X 3 = 9)	Marks

29.	(A)
	def show():
	f=open("Email.txt",'r')
	data=f.read()
	words=data.split()
	for word in words:
	if '@cmail' in word:
	print(word,end=' ')
	f.close()
	(½ mark for correct function header)
	(½ mark for correctly opening the file)
	(½ mark for correctly reading from the file)
	(½ mark for splitting the text into words)
	(1 mark for correctly displaying the desired words)
	OR
	(B)
	def display_long_words():
	with open("Words.txt", 'r') as file:
	data=file.read()
	words=data.split()
	for word in words:
	if len(word)>5:
	print(word,end=' ')
	(½ mark for correct function header)
	(½ mark for correctly opening the file)
	(½ mark for correctly reading from the file)
	(½ mark for splitting the text into words)
	(1 mark for correctly displaying the desired words)
		(3)

30. (A)

(I)

```
def push_book(BooksStack, new_book):
    BooksStack.append(new_book)
```

(II)

```
def pop_book(BooksStack):
    if not BooksStack:
        print("Underflow")
    else:
        return(BookStack.pop())
```

(III)

```
def peep(BooksStack):
    if not BooksStack:
        print("None")
    else:
        print(BookStack[-1])
```

(3x1 mark for correct function body; No marks for any function header as it was a part of the question)

OR

(B)

```
def push_even_numbers(N):
    EvenNumbers = []
    for num in N:
        if num % 2 == 0:
            EvenNumbers.append(num)
    return EvenNumbers


VALUES = []

for i in range(5):
    VALUES.append(int(input("Enter an integer: ")))

EvenNumbers = push_even_numbers(VALUES)

def pop_even():
    if not EvenNumbers:
        print("Underflow")
    else:
        print(EvenNumbers.pop())

pop_even()
```

(3)

	def Disp_even():
	  if not EvenNumbers:
	  print("None")
	  else:
	  print(EvenNumbers[-1])
	Disp_even()
	(1/2 for identifying even numbers)
	(1/2 mark for correctly adding data to stack)
	(1/2 mark for correctly poping data on the stack and 1/2 mark for checking condition)
	(1/2 mark for correctly displaying the data with none)
	(1/2 mark for function call statements)	

31.	(A) 15@ 7@ 9 OR (B) 1 #2 #3# 1 #2 #3 # 1 # *(1 mark for each correct line of output)* *(deduct ½ mark for not printing @/#)*	(3)

Q No.	SECTION D (4 X 4 = 16)	Marks
32.	(A) (I) select Product, sum(Quantity) from orders group by product having sum(Quantity)>=5; (II) select * from orders order by Price desc; (III) select distinct C_Name from orders; (IV) select sum(price) as total_price from orders where Quantity IS NULL; *(4 x 1 mark for each correct query)* **OR** (B) (I) C_Name | Total_Quantity --- |--- Jitendra | 1 Mustafa | 2 Dhwani | 1	(4)

(II)

```
O_Id | C_Name | Product        | Quantity  | Price
-----  |--------     |------------      |----------    |-------
1002 | Mustafa  | Smartphone | 2           | 10000
1003 | Dhwani   | Headphone  | 1           | 1500
```

(III)

```
O_Id | C_Name   | Product        | Quantity | Price
-----       |----------     |------------      |----------    |-------
1001      | Jitendra    | Laptop         | 1           | 12000
1002      | Mustafa    | Smartphone | 2           | 10000
1003      | Dhwani    | Headphone   | 1           | 1500
```

(IV)

```
MAX(Price)
----------
12000
```

(4 x *1 mark for each correct output)*

33. (I)

```
def show():
    import csv
    f=open("happiness.csv",'r')
    records=csv.reader(f)
    next(records, None) #To skip the Header row
    for i in records:
        if int(i[1])>5000000:
            print(i)
    f.close()
```

(½ mark for opening in the file in right mode)
(½ mark for correctly creating the reader object)
(½ mark for correctly checking the condition)
(½ mark for correctly displaying the records)

(4)

(II)

```
def Count_records():
    import csv
    f=open("happiness.csv",'r')
    records=csv.reader(f)
    next(records, None) #To skip the Header row
    count=0
    for i in records:
        count+=1
    print(count)
    f.close()
```

	(½ mark for opening in the file in right mode) *(½ mark for correctly creating the reader object)* *(½ mark for correct use of counter)* *(½ mark for correctly displaying the counter)* **Note (**for both parts (I) and (II)**):** *(i) Ignore* **import csv** *as it may be considered the part of the complete program, and there is no need to import it in individual functions.* *(ii) Ignore next(records, None) as the file may or may not have the Header Row.*	
34.	(I) Select * from FACULTY natural join COURSES where Salary<12000; Or Select * from FACULTY, COURSES where Salary<12000 and facuty.f_id=courses.f_id; (II) Select * from courses where fees between 20000 and 50000; (III) Update courses set fees=fees+500 where CName like '%Computer%'; (IV) (A) Select FName, LName from faculty natural join courses where Came="System Design"; Or Select FName, LName from faculty, courses where Came="System Design" and facuty.f_id=courses.f_id; OR (B) Select * from FACULTY, COURSES; *(4x1 mark for each correct query)*	(4)

35.	def AddAndDisplay():
	import mysql.connector as mycon
	mydb=mycon.connect(host="localhost",user="root",
	passwd="Pencil",database="ITEMDB")
	mycur=mydb.cursor()
	no=int(input("Enter Item Number: "))
	nm=input("Enter Item Name: ")
	pr=float(input("Enter price: "))
	qty=int(input("Enter qty: "))
	query="INSERT INTO stationery VALUES ({},'{}',{},{})"
	query=query.format(no,nm,pr,qty)
	mycur.execute(query)
	mydb.commit()
	mycur.execute("select * from stationery where price>120")
	for rec in mycur:
	print(rec)
		(4)

	(½ mark for correctly importing the connector object) *(½ mark for correctly creating the connection object)* *(½ mark for correctly creating the cursor object)* *(½ mark for correctly inputting the data)* *(½ mark for correct creation of first query)* *(½ mark for correctly executing the first query with commit)* *(½ mark for correctly executing the second query)* *(½ mark for correctly displaying the data)*	

Q No.	SECTION E (2 X 5 = 10)	Marks

36.	(I)
	import pickle
	
	def input_candidates():
	candidates = []
	n = int(input("Enter the number of candidates you want to add: "))
	for i in range(n):
	candidate_id = int(input("Enter Candidate ID: "))
	candidate_name = input("Enter Candidate Name: ")
	designation = input("Enter Designation: ")
	experience = float(input("Enter Experience (in years): "))
	candidates.append([candidate_id, candidate_name, designation, experience])
	return candidates
	candidates_list = input_candidates()
	
	def append_candidate_data(candidates):
	with open('candidates.bin', 'ab') as file:
	for candidate in candidates:
	pickle.dump(candidate, file)
	print("Candidate data appended successfully.")
	
	append_candidate_data(candidates_list)
	
	(II)
	import pickle
	
	def update_senior_manager():
	updated_candidates = []
	with open('candidates.bin', 'rb') as file:
	while True:
	candidate = pickle.load(file)
	if candidate[3] > 10:
	candidate[2] = 'Senior Manager'
	updated_candidates.append(candidate)
	
	with open('candidates.bin', 'wb') as file:
	for candidate in updated_candidates:
		(5)

```
            pickle.dump(candidate, file)

    print("Candidates updated to Senior Manager where applicable.")

update_senior_manager()
```

(III)

```
import pickle

def display_non_senior_managers():
    with open('candidates.bin', 'rb') as file:
        while True:
            candidate = pickle.load(file)
            if candidate[2] != 'Senior Manager':  # Check if not Senior Manager
                print(f"Candidate ID: {candidate[0]}")
                print(f"Candidate Name: {candidate[1]}")
                print(f"Designation: {candidate[2]}")
                print(f"Experience: {candidate[3]}")
                print("--------------------")
display_non_senior_managers()
```

(1/2 mark of import pickle)
(1/2 mark for input)
(1/2 mark for opening file in append mode and 1/2 mark for using dump)
(1/2 mark for opening file in read mode and 1/2 mark for using load)
(1 mark for checking the condition and updating the value)
(1 mark for checking the condition and displaying data correctly)

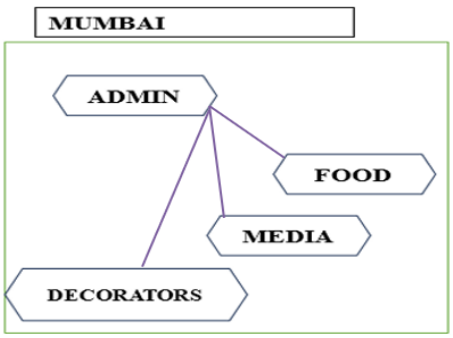
37. (I) ADMIN Block as it has maximum number of computers. (5)

(1 mark for correct answer)

(II) Switch

(1 mark for correct answer)

(III)



(or Any other correct layout)

Cable: Coaxial cable

(½ mark for correct layout + ½ mark for correct table type)

	(IV) There is no requirement of the Repeat as the optical fibre cable used for the network can carry the data to much longer distances than within the campus. *(1 mark for correct answer)* (V) (A) a) Video Conferencing OR (B) LAN *(1 mark for correct answer)*	

SAMPLE QUESTION PAPER (THEORY)
CLASS: XII SESSION: 2024-25
COMPUTER SCIENCE (083)

Time allowed: 3 Hours **Maximum Marks: 70**

General Instructions:

- This question paper contains 37 questions.
- All questions are compulsory. However, internal choices have been provided in some questions. Attempt only one of the choices in such questions
- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A consists of 21 questions (1 to 21). Each question carries 1 Mark.
- Section B consists of 7 questions (22 to 28). Each question carries 2 Marks.
- Section C consists of 3 questions (29 to 31). Each question carries 3 Marks.
- Section D consists of 4 questions (32 to 35). Each question carries 4 Marks.
- Section E consists of 2 questions (36 to 37). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.
- In case of MCQ, text of the correct answer should also be written.

Q No.	Section-A (21 x 1 = 21 Marks)	Marks
1.	State True or False: The Python interpreter handles logical errors during code execution.	(1)

2.	Identify the output of the following code snippet:
	`text = "PYTHONPROGRAM"`
	`text=text.replace('PY','#')`
	`print(text)`
	(A) #THONPROGRAM
	(B) ##THON#ROGRAM
	(C) #THON#ROGRAM
	(D) #YTHON#ROGRAM
		(1)

3.	Which of the following expressions evaluates to False? (A) not(True) and False (B) True or False (C) not(False and True) (D) True and not(False)	(1)

4.	What is the output of the expression?
	`country='International'`
	`print(country.split("n"))`
	(A) ('I', 'ter', 'atio', 'al')
	(B) ['I', 'ter', 'atio', 'al']
	(C) ['I', 'n', 'ter', 'n', 'atio', 'n', 'al']
	(D) Error
		(1)

5.	What will be the output of the following code snippet?
	`message= "World Peace"`
	`print(message[-2::-2])`
		(1)

6.	What will be the output of the following code?
	`tuple1 = (1, 2, 3)`
	`tuple2 = tuple1`
	`tuple1 += (4,)`
	`print(tuple1 == tuple2)`
	(A) True
	(B) False
	(C) tuple1
	(D) Error
		(1)

7.	If my_dict is a dictionary as defined below, then which of the following statements will raise an exception? `my_dict = {'apple': 10, 'banana': 20, 'orange': 30}` (A) my_dict.get('orange') (B) print(my_dict['apple', 'banana']) (C) my_dict['apple']=20 (D) print(str(my_dict))	(1)
8.	What does the list.remove(x) method do in Python? (A) Removes the element at index x from the list (B) Removes the first occurrence of value x from the list (C) Removes all occurrences of value x from the list (D) Removes the last occurrence of value x from the list	(1)
9.	If a table which has one Primary key and two alternate keys. How many Candidate keys will this table have? (A) 1 (B) 2 (C) 3 (D) 4	(1)

10.	Write the missing statement to complete the following code:
	`file = open("example.txt", "r")`
	`data = file.read(100)`
	`_____ #Move the file pointer to the beginning of the file`
	`next_data = file.read(50)`
	`file.close()`
		(1)

11.	State whether the following statement is True or False: The **`finally`** block in Python is executed only if no exception occurs in the **`try`** block.	(1)

12.	What will be the output of the following code?
	`c = 10`
	`def add():`
	`  global c`
	`  c = c + 2`
	`  print(c,end='#')`
	`add()`
	`c=15`
	`print(c,end='%')`
	(A) 12%15#
	(B) 15#12%
	(C) 12#15%
	(D) 12%15#
		(1)

13.	Which SQL command can change the degree of an existing relation?	(1)
14.	What will be the output of the query? `SELECT * FROM products WHERE product_name LIKE 'App%';` (A) Details of all products whose names start with 'App' (B) Details of all products whose names end with 'App' (C) Names of all products whose names start with 'App' (D) Names of all products whose names end with 'App'	(1)
15.	In which datatype the value stored is padded with spaces to fit the specified length. (A) DATE (B) VARCHAR (C) FLOAT (D) CHAR	(1)
16.	Which aggregate function can be used to find the cardinality of a table? (A) sum() (B) count() (C) avg() (D) max()	(1)
17.	Which protocol is used to transfer files over the Internet? (A) HTTP (B) FTP (C) PPP (D) HTTPS	(1)

18.	Which network device is used to connect two networks that use different protocols? (A) Modem (B) Gateway (C) Switch (D) Repeater	(1)
19.	Which switching technique breaks data into smaller packets for transmission, allowing multiple packets to share the same network resources.	(1)
	Q20 and Q21 are Assertion(A) and Reason(R) based questions. Mark the correct choice as: **(A) Both A and R are true and R is the correct explanation for A** **(B) Both A and R are true and R is not the correct explanation for A** **(C) A is True but R is False** **(D) A is False but R is True**	
20.	Assertion (A): Positional arguments in Python functions must be passed in the exact order in which they are defined in the function signature. Reasoning (R): This is because Python functions automatically assign default values to positional arguments.	(1)
21.	Assertion (A): A SELECT command in SQL can have both WHERE and HAVING clauses. Reasoning (R): WHERE and HAVING clauses are used to check conditions, therefore, these can be used interchangeably.	(1)

Q No	Section-B (7 x 2=14 Marks)	Marks
22.	How is a mutable object different from an immutable object in Python? Identify one mutable object and one immutable object from the following: (1,2), [1,2], {1:1,2:2}, '123'	(2)
23.	Give two examples of each of the following: (I) Arithmetic operators (II) Relational operators	(2)
24.	If L1=[1,2,3,2,1,2,4,2, . . .], and L2=[10,20,30, . . .], then *(Answer using builtin functions only)* (I) A) Write a statement to count the occurrences of 4 in L1. OR B) Write a statement to sort the elements of list L1 in ascending order.	(2)

	(II) A) Write a statement to insert all the elements of L2 at the end of L1. OR B) Write a statement to reverse the elements of list L2.	

25.	Identify the correct output(s) of the following code. Also write the minimum and the maximum possible values of the variable b.
	`import random`
	`a="Wisdom"`
	`b=random.randint(1,6)`
	`for i in range(0,b,2):`
	`  print(a[i],end='#')`
	(A) W# (B) W#i#
	(C) W#s# (D) W#i#s#
		(2)

26.	The code provided below is intended to swap the first and last elements of a given tuple. However, there are syntax and logical errors in the code. Rewrite it after removing all errors. Underline all the corrections made.
	`def swap_first_last(tup)`
	`  if len(tup) < 2:`
	`  return tup`
	`  new_tup = (tup[-1],) + tup[1:-1] + (tup[0])`
	`  return new_tup`
	`result = swap_first_last((1, 2, 3, 4))`
	`print("Swapped tuple: "  result)`
		(2)

27.	(I) A) What constraint should be applied on a table column so that duplicate values are not allowed in that column, but NULL is allowed. **OR** B) What constraint should be applied on a table column so that NULL is not allowed in that column, but duplicate values are allowed.	(2)

	(II) A) Write an SQL command to remove the Primary Key constraint from a table, named MOBILE. M_ID is the primary key of the table. OR B) Write an SQL command to make the column M_ID the Primary Key of an already existing table, named MOBILE.	
28.	A) List one advantage and one disadvantage of star topology. **OR** B) Expand the term SMTP. What is the use of SMTP?	(2)

Q No.	Section-C (3 x 3 = 9 Marks)	Marks
29.	A) Write a Python function that displays all the words containing **@cmail** from a text file "Emails.txt". **OR** B) Write a Python function that finds and displays all the words longer than 5 characters from a text file "Words.txt".	(3)
30.	A) You have a stack named **BooksStack** that contains records of books. Each book record is represented as a list containing **book_title**, **author_name**, and **publication_year**. Write the following user-defined functions in Python to perform the specified operations on the stack **BooksStack**: (I) push_book(BooksStack, new_book): This function takes the stack BooksStack and a new book record new_book as arguments and pushes the new book record onto the stack. (II) pop_book(BooksStack): This function pops the topmost book record from the stack and returns it. If the stack is already empty, the function should display "Underflow". (III) peep(BookStack): This function displays the topmost element of the stack without deleting it. If the stack is empty, the function should display 'None'. **OR** (B) Write the definition of a user-defined function `push_even(N)` which accepts a list of integers in a parameter `N` and pushes all those integers which are even from the list `N` into a Stack named `EvenNumbers`. Write function pop_even() to pop the topmost number from the stack and returns it. If the stack is already empty, the function should display "Empty". Write function Disp_even() to display all element of the stack without deleting them. If the stack is empty, the function should display 'None'.	(3)

For example:
If the integers input into the list `VALUES` are:
[10, 5, 8, 3, 12]

Then the stack `EvenNumbers` should store:
[10, 8, 12]

31. Predict the output of the following code: (3)

```
d = {"apple": 15, "banana": 7, "cherry": 9}
str1 = ""
for key in d:
    str1 = str1 + str(d[key]) + "@" + "\n"
str2 = str1[:-1]
print(str2)
```

OR

Predict the output of the following code:

```
line=[4,9,12,6,20]
for I in line:
   for j in range(1,i%5):
       print(j,'#',end="")
   print()
```

Q No.	Section-D (4 x 4 = 16 Marks)	Marks

32. Consider the table ORDERS as given below (4)

O_Id	C_Name	Product	Quantity	Price
1001	Jitendra	Laptop	1	12000
1002	Mustafa	Smartphone	2	10000
1003	Dhwani	Headphone	1	1500

Note: The table contains many more records than shown here.

A) Write the following queries:

(I) To display the total Quantity for each Product, excluding Products with total Quantity less than 5.

(II) To display the orders table sorted by total price in descending order.

(III) To display the distinct customer names from the Orders table.

(IV) Display the sum of Price of all the orders for which the quantity is null.

OR

B) Write the output

```
(I) Select c_name, sum(quantity) as total_quantity
from orders group by c_name;
```

```
(II) Select * from orders where product like
'%phone%';
```

```
(III) Select o_id, c_name, product, quantity, price
from orders where price between 1500 and 12000;
```

```
(IV) Select max(price) from orders;
```

33. A csv file "Happiness.csv" contains the data of a survey. Each record of the file contains the following data: (4)

- Name of a country
- Population of the country
- Sample Size (*Number of persons who participated in the survey in that country*)
- Happy (*Number of persons who accepted that they were Happy*)

For example, a sample record of the file may be:

['Signiland', 5673000, 5000, 3426]

Write the following Python functions to perform the specified operations on this file:

(I) Read all the data from the file in the form of a list and display all those records for which the population is more than 5000000.

(II) Count the number of records in the file.

34. Saman has been entrusted with the management of Law University Database. He needs to access some information from FACULTY and COURSES tables for a survey analysis. Help him extract the following information by writing the desired SQL queries as mentioned below. (4)

Table: **FACULTY**

F_ID	FName	LName	Hire_Date	Salary
102	Amit	Mishra	12-10-1998	12000
103	Nitin	Vyas	24-12-1994	8000
104	Rakshit	Soni	18-5-2001	14000
105	Rashmi	Malhotra	11-9-2004	11000
106	Sulekha	Srivastava	5-6-2006	10000

Table: **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 Basic	6000

(I) To display complete details (from both the tables) of those Faculties whose salary is less than 12000.

(II) To display the details of courses whose fees is in the range of 20000 to 50000 (both values included).

(III) To increase the fees of all courses by 500 which have "Computer" in their Course names.

(IV) (A) To display names (FName and LName) of faculty taking System Design.

OR

(B) To display the Cartesian Product of these two tables.

35. A table, named STATIONERY, in ITEMDB database, has the following structure: (4)

Field	Type
itemNo	int(11)
itemName	varchar(15)
price	float
qty	int(11)

Write the following Python function to perform the specified operation:
AddAndDisplay(): To input details of an item and store it in the table STATIONERY. The function should then retrieve and display all records from the STATIONERY table where the Price is greater than 120.

Assume the following for Python-Database connectivity:
Host: localhost, User: root, Password: Pencil

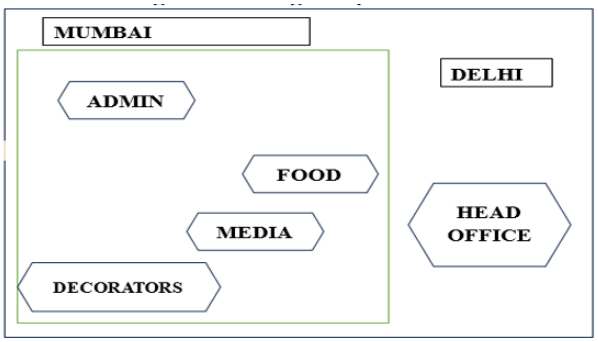
Q.No.	SECTION E (2 X 5 = 10 Marks)	Marks

36. Surya is a manager working in a recruitment agency. He needs to manage the records of various candidates. For this, he wants the following information of each candidate to be stored: (5)
- Candidate_ID – integer
- Candidate_Name – string
- Designation – string
- Experience – float

You, as a programmer of the company, have been assigned to do this job for Surya.

(I) Write a function to input the data of a candidate and append it in a binary file.

	(II) Write a function to update the data of candidates whose experience is more than 10 years and change their designation to "Senior Manager". (III) Write a function to read the data from the binary file and display the data of all those candidates who are not "Senior Manager".	
37.	Event Horizon Enterprises is an event planning organization. It is planning to set up its India campus in Mumbai with its head office in Delhi. The Mumbai campus will have four blocks/buildings - ADMIN, FOOD, MEDIA, DECORATORS. You, as a network expert, need to suggest the best network-related solutions for them to resolve the issues/problems mentioned in points (I) to (V), keeping in mind the distances between various blocks/buildings and other given parameters.	(5)



Block to Block distances (in Mtrs.)

From	To	Distance
ADMIN	FOOD	42 m
ADMIN	MEDIA	96 m
ADMIN	DECORATORS	48 m
FOOD	MEDIA	58 m
FOOD	DECORATORS	46 m
MEDIA	DECORATORS	42 m

Distance of Delhi Head Office from Mumbai Campus = 1500 km
Number of computers in each of the blocks/Center is as follows:

ADMIN	30
FOOD	18
MEDIA	25
DECORATORS	20
DELHI HEAD OFFICE	18

	(I) Suggest the most appropriate location of the server inside the MUMBAI campus. Justify your choice. (II) Which hardware device will you suggest to connect all the computers within each building? (III) Draw the cable layout to efficiently connect various buildings within the MUMBAI campus. Which cable would you suggest for the most efficient data transfer over the network? (IV) Is there a requirement of a repeater in the given cable layout? Why/ Why not? (V) A) What would be your recommendation for enabling live visual communication between the Admin Office at the Mumbai campus and the DELHI Head Office from the following options: a) Video Conferencing b) Email c) Telephony d) Instant Messaging **OR** B) What type of network (PAN, LAN, MAN, or WAN) will be set up among the computers connected in the MUMBAI campus?	