Kendriya Vidyalaya Sangathan, Jaipur Region
Pre-Board Examination: 2024-25
Marking scheme Set No: 1

Class: XII **Subject**: Computer Science (083)

Maximum Marks: 70 **Period**: 3 Hours

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 the case of MCQ, the text of the correct answer should also be written.

Q.	Section-A (21 x 1 = 21 Marks)	Mark
Ans 1	True (1 mark for correct answer)	1
Ans 2	(c) KENDRI* VID*LA* (1 mark for correct answer)	1
Ans 3	(b) False (1 mark for correct answer)	1
Ans 4	(c) [“abc”, “pink”, “city”] (1 mark for correct answer)	1
Ans 5	tyJvmya (1 mark for correct answer)	1
Ans 6	(c) tp1 = tp+tp (1 mark for correct answer)	1
Ans 7	(b) print(my_dict['aman', 'sumit']) (1 mark for correct answer)	1
Ans 8	(a) pop( ) (1 mark for correct answer)	1
Ans 9	(c) Roll No (1 mark for correct answer)	1
Ans10	file.seek(0) (OR file.seek(0,0)) (1 mark for correct answer)	1
Ans11	True (1 mark for correct answer)	1
Ans12	(a) 20#50@20 (1 mark for correct answer)	1
Ans13	(c) UPDATE (1 mark for correct answer)	1
Ans14	(b) SELECT * FROM Item ORDER BY IName DESC ; (1 mark for correct answer)	1
Ans15	(a) For pattern matching (1 mark for correct answer)	1
Ans16	(c) ALL the values (1 mark for correct answer)	1
Ans17	(c) Hypertext transfer protocol (1 mark for correct answer)	1

Ans18	(c) Gateway (1 mark for correct answer)	1
Ans19	Message Switching technique (1 mark for correct answer)	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	
Ans20	(b) Both A and R are true and R is not the correct explanation for A (1 mark for correct answer)	1
Ans21	(c) A is True but R is False (1 mark for correct answer)	

Q	**Section-B (7 x 2=14 Marks)**	Mark
Ans 22	Immutable types are those that can never change their value in place. These are : integers, float, string, tuple Mutable types are those whose values can be changed in place. These are : lists, dictionaries (1 mark for each correct definition and example)	2
Ans 23	(i) True (ii) 2 (1 mark for each correct answer.)	2
Ans 24	(I) A) L1.remove(4) OR B) L1.sort( ) (II) A) L1.insert( ) OR B) L1.extend(L2) (1 mark for each correct answer)	2
Ans 25	(ii)30#40#50# Maximum value assigned to FROM is 3 Maximum value assigned to TO is 4 (1 mark for each correct answer) (½ x 2 = 1 Mark for correct Maximum values)	2

Ans 26	p=30
	for c in range(0,p):
	if c%4==0:
	print (c*4)
	elif c%5==0:
	print (c+3)
	else:
	print(c+10)
	(½ mark each for correcting 4 mistakes)
		2

Ans 27	(I) A) UNIQUE OR B) NOT NULL (1 mark for correct answer) (II) A) ALTER TABLE MOBILE DROP PRIMARY KEY; OR	2

B) ALTER TABLE MOBILE ADD PRIMARY KEY (M_ID);
(1 mark for correct answer)

28 Ans: Fault diagnosis is easier in Star topology as if there is any problem in a node will affect the particular node only. While in bus topology, if problem exists in common medium it will affect the entire nodes. 2

OR

Web Server	Web browser
A web server is a computer or a group of computers that hosts or stores content of website. It processes and delivers web pages of the websites to the users. The main job of a web server is to display the website content	A web browser is an application used to access and view websites. Common web browsers include Microsoft Internet Explores, Google Chrome etc.

(2 mark for correct answer)

Q	Section-C (3 x 3 = 9 Marks)	Mark

29 (A) 3

Logic:- We need to split all the words of a every line and then check first word of each line.

```
def count_the():
       F=open("abc.txt","r")
       L=F.readlines()
       count=0
       for x in L: # read all the lines one by one
              L1=x.split() # It will split all the lines into words
              if L1[0].lower() == "the" :
                      count=count+1
        print("total lines are :- ",count)
      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/character)
(1 mark for correctly displaying the desired counts)

OR

(B)

Read the file character by character

```
def count_vowels():
       F=open("abc.txt", "r")
       S=f.read()
       Count=0
       for x in S:
             if x.lower() in "aeiou":
                    count=count+1
        print("total number of vowels are :- ", count)
      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 checking vowels)
(1 mark for correctly displaying the desired counts)

30 N=[12, 13, 34, 56, 21, 79, 98, 22,35, 38] 3

```
def PUSH(S,N):
    S.append(N)
def POP(S):
    if S!=[ ]:
        return S.pop()
    else:
        return None
ST=[ ]
for k in N:
    if k%2!=0:
        PUSH(ST,k)
    while True:
        if ST!=[ ]:
            print(POP(ST),end=" ")
        else:
            break
```

(2x1 mark for correct function body; 1 mark for function calling)

OR

```
(B) N= [2,5,10,13,20,23,45,56,60,78]
def PUSH(S,N):
    S.append(N)
def POP(S):
    if S!=[]:
        return S.pop()
    else:
        return None
ST=[]
for k in N:
    if k%5==0:
        PUSH(ST,k)
    while True:
        if ST!=[]:
            print(POP(ST),end=" ")
        else:
            break
```

(2x1 mark for correct function body; 1 mark for function calling)

31	2 4 6 5 OR 47.0 35.0 54.0 26.0 (3 Mark for correct output, 1 mark for partial correct)	3

Q	Section-D (4 x 4 = 16 Marks)	Mark
32	i) Select ActivityName, Acode from ACTIVITY order by Acode desc; ii) Select sum(PrizeMoney) from ACTIVITY group by ParticipantsNum; iii) Select ScheduleDate, ParticipantsNum from ACTIVITY where ActivityName = ‘Relay 100 x4’; iv) Update ACTIVITY set PrizeMoney = PrizeMoney + 500 where ActivityName = “High jump”; (1 mark for each correct Query.) OR	4

(i)

Count(distinct ParticipantsNum)
3

(ii)

Max(ScheduleDate)	Min(ScheduleDate)
2004-03-19	2003-12-12

(iii)

Sum(PrizeMoney)
54000

(iv) delete from ACTIVITY where Acode = 1003;

(1 mark for each correct answer.)

Ans 33

4

Ans: (a)

```
import csv
def AddNewRec(Country,Capital):
        f=open("CAPITAL.CSV",'a')
        fwriter=csv.writer(f,lineterminator="\n")
        fwriter.writerow([Country,Capital])
        f.close()
```

(b)

```
def ShowRec():
        with open("CAPITAL.CSV","r") as NF:
        NewReader=csv.reader(NF)
        for rec in NewReader:
              print(rec[0],rec[1])

AddNewRec("FRANCE","PARIS")
AddNewRec("SRILANKA","COLOMBO")
ShowRec()
```

Ans 34

4

SQL Two table based question

(i) SELECT CNAME, PRICE FROM COMPANY, CUSTOMER WHERE COMPANY.CID = CUSTOMER.CID AND PRICE < 30000;

(ii) SELECT CNAME, PRICE FROM COMPANY, CUSTOMER WHERE COMPANY.CID = CUSTOMER.CID AND PRICE BETWEEN 20000 AND 35000;

(iii) UPDATE CUSTOMER SET PRICE = PRICE+1000 WHERE NAME LIKE “S%”;

(iv) SELECT PRODUCTNAME,CITY, PRICE FROM COMPANY,CUSTOMER WHERE COMPANY.CID=CUSTOMER.CID AND PRODUCTNAME=”MOBILE”;

(1 mark for each correct query.)

Ans 35

4

```
import mysql.connector as mysql
con1 = mysql.connect(host = “localhost”, user = “root”, password = “tiger”,
               database = “sample”)
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( )
```

(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) (1 mark for correct creation of query) (½ mark for correctly executing the first query with commit (½ mark for correctly displaying the data)	

Q	Section-E (2 x 5 = 10 Marks)	Mark
Ans 36	(I) import pickle	5

```
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 = [ ]
    try:
       with open('candidates.bin', 'rb') as file:
          while True:
              try:
                 candidate = pickle.load(file)
                 if candidate[3] > 10: # If experience > 10 years
                    candidate[2] = 'Senior Manager'
                 updated_candidates.append(candidate)
               except EOFError:
        break # End of file reached
     except FileNotFoundError:
        print("No candidate data found. Please add candidates first.")
        return
     with open('candidates.bin', 'wb') as file:
        for candidate in updated_candidates:
            pickle.dump(candidate, file)
     print("Candidates updated to Senior Manager where applicable.")
update_senior_manager()
```

(III)

```
import pickle
def display_non_senior_managers():
   try:
        with open('candidates.bin', 'rb') as file:
            while True:
                try:
                    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("--------------------")
                except EOFError:
                    break # End of file reached
    except FileNotFoundError:
        print("No candidate data found. Please add candidates first.")
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)
(1 mark for try and except block)

37	Networking based question (i) Server should be installed in Building 3 , As maximum computer is there (ii) Hub/Switch installed in each and every building to connect computers in each building. (iii) Bus/ Star topology (iv) No repeater is required as distance between building is less than 100 meter. (v) Voip OR LAN (Local Area Network) (1x5=5 mark for each correct answer)	5