

KENDRIYA VIDYALAYA SANGATHAN
GUWAHATI REGION
PRE BOARD – II EXAMINATION (Session 2023-24)

Class : XII
Subject: (083) Computer Science

Time Allowed: 03:00 Hours
Maximum Marks : 70

General Instructions:

- Please check this question paper contains 35 questions.
- The paper is divided into 4 Sections- A, B, C, D and E.
- Section A, consists of 18 questions (1 to 18). Each question carries 1 Mark.
- Section B, consists of 7 questions (19 to 25). Each question carries 2 Marks.
- Section C, consists of 5 questions (26 to 30). Each question carries 3 Marks.
- Section D, consists of 2 questions (31 to 32). Each question carries 4 Marks.
- Section E, consists of 3 questions (33 to 35). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.

SECTION – A		
Q1.	True	(1)
Q2.	C) A view of existing column with different name	(1)
Q3.	c) 512	(1)
Q4.	c) {'A':4000, 'B':2500, 'C':3000}	(1)
Q5.	b) 8,15	(1)
Q6.	b) Router	(1)
Q7.	(b) a dictionary is a set of key-value pairs	(1)
Q8.	b)Ye-r 2024 -ll the best	(1)
Q9.	d) Statement 4	(1)
Q10.	a) rgb	(1)
Q11.	c) distinct()	(1)
Q12.	c) ['q']#['u']#['i']#['t']#['e']#	(1)
Q13.	a)cursor.rowcount	(1)
Q14.	a) Aggregate functions ignore NULL	(1)
Q15.	a. TCP	(1)
Q16.	b) SEEK	(1)
Q17.	a)	(1)
Q18.	d)	(1)

SECTION – B

Q19.	<p>(i) ½ mark for each correct expansion</p> <p>HTML(Hypertext mark Up language)</p> <ul style="list-style-type: none"> ☒ 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 sensitive <p>XML (Extensible Markup Language)</p> <ul style="list-style-type: none"> ☒ we can define our own tags and use them ☒ Dynamic web development language – as it is used for transporting and storing data ☒ Case sensitive <p>1 mark for any one correct difference No mark to be awarded if only full form is given</p> <p style="text-align: center;">OR</p> <p>(I) 1 mark for correct definition</p> <p>(ii) 1/2 mark for each correct name of web browser.</p>	(2)
Q20.	<pre> Def checkNumber(N): status = N%2 return #main-code num=int(input(" Enter a number to check :)) mark k=checkNumber(num) if k = 0: print("This is EVEN number") else: print("This is ODD number") </pre> <p># Def should be def</p> <p># return what? Should be return status</p> <p># Message not enclosed within quotation</p> <p># must be k = = 0</p> <p>(½ mark for each correct correction made and underlined.)</p>	(2)
Q21.	<p>½ mark for correct function header</p> <p>½ mark for correct loop</p> <p>½ mark for correct if statement</p> <p>½ mark for displaying the output</p> <p style="text-align: center;">OR</p> <p>½ mark for correct function header</p> <p>½ mark for correct loop</p> <p>½ mark for correct if statement</p> <p>½ mark for displaying the output</p>	(2)
Q22.	<p>unchanged (30, 50)</p> <p>changed (16, 50)</p> <p>unchanged (16, 22)</p> <p>changed (11, 22)</p> <p>(½ mark for each correct Output)</p>	(2)

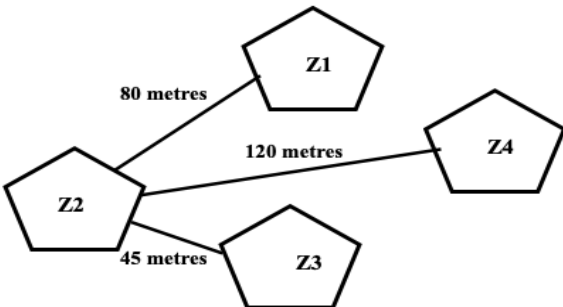
Q23.	<p>d) All are possible OR Minimum 0 and maximum 1 (1 mark for each correct Output)</p>	(2)																		
Q24.	<p>1 mark for correct ALTER TABLE command SQL Command to add primary key: ALTER TABLE Student ADD Stuid INTEGER PRIMARY KEY; 1 mark for correct INSERT command As the primary key is added as the last field, the command for inserting data will be: INSERT INTO Student VALUES("Shweta","XII",98,1299); Alternative answer: INSERT INTO Student(Stuid,Sname,class,Mark) VALUES(1299,"Shweta","XII",98);</p>	(2)																		
Q25.	<p>50#5 (2 marks for correct answer) OR 20 40 (2 marks for correct answer)</p>	(2)																		
SECTION – C																				
Q26.	<p>ND-*34 (½ mark for each correct character)</p>	(3)																		
Q27.	<p>1 mark for each correct output (i)</p> <table border="1" data-bbox="170 1050 500 1270"> <tr><td>MAKE</td></tr> <tr><td>Toyota</td></tr> <tr><td>Tata</td></tr> <tr><td>Renault</td></tr> <tr><td>Suzuki</td></tr> </table> <p>(ii)</p> <table border="1" data-bbox="170 1329 690 1549"> <thead> <tr> <th>MAKE</th> <th>COUNT(*)</th> </tr> </thead> <tbody> <tr><td>Toyota</td><td>1</td></tr> <tr><td>Tata</td><td>2</td></tr> <tr><td>Renault</td><td>2</td></tr> <tr><td>Suzuki</td><td>1</td></tr> </tbody> </table> <p>(iii)</p> <table border="1" data-bbox="170 1608 482 1745"> <tr><td>CNAME</td></tr> <tr><td>Duster</td></tr> <tr><td>Fortuner</td></tr> </table>	MAKE	Toyota	Tata	Renault	Suzuki	MAKE	COUNT(*)	Toyota	1	Tata	2	Renault	2	Suzuki	1	CNAME	Duster	Fortuner	(3)
MAKE																				
Toyota																				
Tata																				
Renault																				
Suzuki																				
MAKE	COUNT(*)																			
Toyota	1																			
Tata	2																			
Renault	2																			
Suzuki	1																			
CNAME																				
Duster																				
Fortuner																				
Q28.	<pre>def rdlines(): file = open('visitors.txt','r') for line in file: if line[0] == 'P': print(line) file.close()</pre>	(3)																		

	<p># Call the rdlines function. rdlines() ½ mark for function header 1 mark for opening file 1 mark for correct for loop and condition ½ mark for closing file</p> <p style="text-align: center;">OR</p> <pre>def count_word(): file = open('india.txt','r') count = 0 for line in file: words = line.split() for word in words: if word == 'India': count += 1 print(count) file.close() # call the function count_word(). count_word()</pre> <p>½ mark for function header 1 mark for opening file 1 mark for correct for loop and condition ½ mark for closing file</p>	
Q29.	<p>Select company, avg(Price) from toys group by company having Qty>15; Select Company, count(distinct name) from toys group by Company; Select name, sum(Price* Qty) from toys; ½ mark for the Select with avg(), ½ mark for the having clause ½ mark for the Select with count() , ½ mark for group by clause ½ mark for the Select with sum() , ½ mark for the group by clause</p>	(3)
Q30.	<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> <p>1 ½ marks for each function</p>	(3)

SECTION – D

Q31.	<p>i) SELECT * FROM Consumer ORDER BY ConsumerName DESC (ii) SELECT StationaryName, Price FROM Stationary WHERE Price>=10 AND Price<=15 (iii) SELECT C.ConsumerName, C.City, S.StationaryName FROM Stationary S, Consumer C WHERE C.S_ID=S.S+ID AND S.Company="Reynolds"; iv) UPDATE Stationary SET Price=Price+2</p>	(4)
Q32.	<p>(a) Comma separated value 1 marks (b) (i) 2 Marks <pre>def InsertRow(): import csv f=open("class.csv","a+",newline="") rno=int(input("Enter roll no. :")) name=int(input("Enter name :")) marks=int(input("Enter marks :")) wo=csv.writer(f) wo.writerow([rno, name, marks]) f.close()</pre> <p>(b)(ii) 2 Marks <pre>def COUNTD(): import csv count=0 f=open("class.csv","r") ro=csv.reader(f) for i in ro: if i[2]>75: count+=1 return count</pre> <p>½ mark for opening and closing file ½ mark for reader object ½ mark for print heading ½ mark for printing data</p> </p></p>	(4)

SECTION – E

Q33.	<p>i. Z2 as it has maximum number of computers. ii. For very fast and efficient connections between various blocks within the campus suitable topology: Star Topology</p>  <pre> graph TD Z2 --- 80 metres Z1 Z2 --- 45 metres Z3 Z2 --- 120 metres Z4 </pre>	(5)
------	--	-----

	<p>iii. Repeater: To be placed between Block Z2 to Z4 as distance between them is more than 100 metres.</p> <p>Hub/Switch: To be placed in each block as each block has many computers that needs to be included to form a network.</p> <p>iv. Voice Over Internet Protocol</p> <p>v. WAN as distance between Delhi and Mumbai is more than 40kms.</p> <p>(1 mark for each correct answer)</p>	
34.	<pre> import pickle def createFile(): fobj=open("Book.dat","ab") BookNo=int(input("Book Number : ")) Book_name=input("Name :") Author = input("Author:") Price = int(input("Price : ")) rec=[BookNo,Book_Name,Author,Price] pickle.dump(rec,fobj) fobj.close() def CountRec(Author): fobj=open("Book.dat","rb") num = 0 try: while True: rec=pickle.load(fobj) if Author==rec[2]: num = num + 1 except: fobj.close() return num or import pickle def CountRec(): fobj=open("STUDENT.DAT","rb") num = 0 try: while True: rec=pickle.load(fobj) if rec[2] > 75: print(rec[0],rec[1],rec[2],sep="\t") num = num + 1 except: fobj.close() return num </pre>	5

<p>Q35.</p>	<p>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</p> <p>½ mark for correct definition</p> <p>½ mark for correct example</p> <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> <p>½ mark for importing correct module</p> <p>1 mark for correct connect()</p> <p>½ mark for correctly accepting the input</p> <p>1 ½ mark for correctly executing the query</p> <p>½ mark for correctly using commit()</p> <p>OR</p> <p>(I) 1 mark for correct difference</p> <p>(II)</p> <pre>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()</pre> <p>½ mark for importing correct module</p> <p>1 mark for correct connect()</p> <p>1 mark for correctly executing the query</p> <p>½ mark for correctly using fetchall()</p> <p>1 mark for correctly</p>	<p>(5)</p>
-------------	---	------------