

**KENDRIYA VIDYALAYA SANGATHAN, TINSUKIA REGION**  
**PRE-BOARD I (2022 - 23)**

**Class: XII**

**Subject: COMPUTER SCIENCE (083)(Theory)**

**Max Marks: 70**

**Time:3hrs**

**MARKING SCHEME**

Question No.	SECTION – A (1 Marks)		
1	(d) error	1	
2	(a)1*3	1	
3	(c) None	1	
4	(b) False	1	
5	(a)['Question paper ', '0', '', '-', '3']	1	
6	(c) w+	1	
7	(b) DELETE	1	
8	(c) USE SCHOOL;	1	
9	(a)[1,2]		
10	(a) PRIMARY, CANDIDATE	1	
11	(b) It returns the byte position of the file pointer as an integer.	1	
12	(b) 2	1	
13	(b) POP-POST OFFICE PROTOCOL	1	
14	(c) (10)	1	
15	(b) multiple row functions	1	
16	(d) username, password, hostname, database name	1	
17	(a)Both A and R are true and R is the correct explanation for A	1	
18	(c)A is True but R is False	1	
	<b>SECTION B (2 Marks)</b>		
19	<pre>Value = 30 for VAL in range(0,Value) :           # Error 1     if val%4==0:                       # Error 2         print (VAL*4)     elif val%5==0:                     # Error 3         print (VAL+3)     else:                               # Error 4         print(VAL+10)</pre>	2	
20	<b>BUS TOPOLOGY</b>	<b>STAR TOPOLOGY</b>	2
	Bus topology is a topology where each device is connected to a single cable which is known as the backbone.	Star topology is a topology in which all devices are connected to a central hub	
	In a Bus topology, the failure of the network cable will cause the whole network to fail.	In star topology, if the central hub fails then the whole network fails.	
	In a bus topology, there is a linear arrangement of nodes in a network.	In star topology, there is a non-linear arrangement of nodes in a network.	
<b>OR</b>			

	<b>DOMAIN OF COMPARISON</b>	<b>CIRCUIT SWITCHING</b>	<b>PACKET SWITCHING</b>	
	<b>Definition</b>	Transmission of data is done through a physically mapped circuit between the source and the target.	Data is transmitted in the form of segments called data packets through dynamic channels of data transmission.	
	<b>Data processing</b>	Transmitted data is demodulated and processed at the receiver end through hardware.	The data is processed at the source before transmission as well as at the nodes and receiver end during transmission and post completion respectively.	
	<b>Flexibility</b>	The route of data transmission is predefined and is rigid.	As there is no fixed route of data transmission hence there is flexibility in transmitting packets through different channels as per traffic or other constraints.	
	<b>Utility</b>	Finds maximum usage in voice-over or telephonic communication.	It has wide utility in the field of data transmission over networks.	
21	(a) h dWTop (b) dict_items([('month', 'JANUARY'), ('exam', 'PREBOARD1'), ('EXAM', 'PRE2')])			2
22	<b>CHAR</b>	<b>VARCHAR</b>		2
	CHAR datatype is used to store character strings of fixed length	VARCHAR datatype is used to store character strings of variable length		
	In CHAR, If the length of the string is less than set or fixed-length then it is padded with extra memory space.	In VARCHAR, If the length of the string is less than the set or fixed-length then it will store as it is without padded with extra memory spaces.		
	CHAR stands for “Character”	VARCHAR stands for “Variable Character”		
	Storage size of CHAR datatypes is equal to n bytes i.e. set length	The storage size of the VARCHAR datatype is equal to the actual length of the entered string in bytes.		
	We should use the CHAR datatype when we expect the data values in a column are of the same length.	We should use the VARCHAR datatype when we expect the data values in a column are of variable length.		
	CHAR takes 1 byte for each character	VARCHAR takes 1 byte for each character and some extra bytes for holding length information		
	Better performance than VARCHAR	Performance is not good as compared to CHAR		
23	POP-Post Office Protocol HTTPS-HyperText Transfer Protocol Secure OR A URL (Uniform Resource Locator) is a unique identifier used to locate a resource on the Internet. It is also referred to as a web address. URLs consist of multiple parts - including a protocol and domain name -- that tell a web browser how and where to retrieve a resource.			2
24	[4,6,7,1,6,9,4] [4, 7, 7, 1, 7, 9, 4] [4, 7, 7, 1, 7, 9, 4]  OR (18, 36, 54)			2

25	<table border="1"> <thead> <tr> <th data-bbox="212 78 754 114">Where Clause in SQL</th> <th data-bbox="762 78 1353 114">Having Clause in SQL</th> </tr> </thead> <tbody> <tr> <td data-bbox="212 114 754 181">Filter table based data catering to specific condition</td> <td data-bbox="762 114 1353 181">Group based data under set condition</td> </tr> <tr> <td data-bbox="212 181 754 217">Applicable without GROUP BY clause</td> <td data-bbox="762 181 1353 217">Does not function without GROUP BY clause</td> </tr> <tr> <td data-bbox="212 217 754 253">Row functions</td> <td data-bbox="762 217 1353 253">Column functions</td> </tr> <tr> <td data-bbox="212 253 754 288">Select, update and delete statements</td> <td data-bbox="762 253 1353 288">Only select statement</td> </tr> <tr> <td data-bbox="212 288 754 324">Applied before GROUP BY clause</td> <td data-bbox="762 288 1353 324">Used after GROUP BY clause</td> </tr> <tr> <td data-bbox="212 324 754 392">Used with single row operations such as Upper, Lower and so on</td> <td data-bbox="762 324 1353 392">Applicable with multiple row functions such as Sum, count and so on</td> </tr> </tbody> </table> <p data-bbox="691 392 735 423" style="text-align: center;">OR</p> <p data-bbox="204 423 1417 495">COUNT(*) will count all the rows in the table, including NULL values. On the other hand, COUNT(column name) will count all the rows in the specified column while excluding NULL values.</p>	Where Clause in SQL	Having Clause in SQL	Filter table based data catering to specific condition	Group based data under set condition	Applicable without GROUP BY clause	Does not function without GROUP BY clause	Row functions	Column functions	Select, update and delete statements	Only select statement	Applied before GROUP BY clause	Used after GROUP BY clause	Used with single row operations such as Upper, Lower and so on	Applicable with multiple row functions such as Sum, count and so on	2
Where Clause in SQL	Having Clause in SQL															
Filter table based data catering to specific condition	Group based data under set condition															
Applicable without GROUP BY clause	Does not function without GROUP BY clause															
Row functions	Column functions															
Select, update and delete statements	Only select statement															
Applied before GROUP BY clause	Used after GROUP BY clause															
Used with single row operations such as Upper, Lower and so on	Applicable with multiple row functions such as Sum, count and so on															
<b>SECTION C</b> <b>(3 Marks)</b>																
26	<p data-bbox="252 607 1129 642">(a) ALTER TABLE PRODUCT ADD TOTAL PRICE NUMBER (10,2).</p> <p data-bbox="252 642 296 678">(b)</p> <p data-bbox="331 678 360 714">(i)</p> <table border="1" data-bbox="368 703 619 741"> <tr> <td data-bbox="368 703 571 741">VICE PRINCIPAL</td> <td data-bbox="571 703 619 741">01</td> </tr> </table> <p data-bbox="331 741 360 777">(ii)</p> <table border="1" data-bbox="368 777 416 815"> <tr> <td data-bbox="368 777 416 815">16</td> </tr> </table> <p data-bbox="331 815 360 851">(iii)</p> <table border="1" data-bbox="368 851 536 920"> <tr> <td data-bbox="368 851 536 887">UMESH</td> </tr> <tr> <td data-bbox="368 887 536 920">YASH RAJ</td> </tr> </table> <p data-bbox="331 920 360 956">(iv)</p> <table border="1" data-bbox="368 956 536 1025"> <tr> <td data-bbox="368 956 536 992">5 MALE</td> </tr> <tr> <td data-bbox="368 992 536 1025">2 FEMALE</td> </tr> </table>	VICE PRINCIPAL	01	16	UMESH	YASH RAJ	5 MALE	2 FEMALE	1+2							
VICE PRINCIPAL	01															
16																
UMESH																
YASH RAJ																
5 MALE																
2 FEMALE																
27	<p data-bbox="204 1025 248 1061">(a)</p> <pre data-bbox="204 1061 1161 1352">fname = "python.txt" num_words = 0 f= open(fname, 'r') words = f.read().split()     for a in words:         if (a.lower() == "to" or a.lower() == "the" ):             num_words = num_words + 1 print("Number of words:", num_words) f.close()</pre> <p data-bbox="651 1352 695 1388" style="text-align: center;">OR</p> <p data-bbox="204 1388 248 1424">(b)</p> <pre data-bbox="204 1424 786 1715">fh=open("python.txt","r")     fw=open("python1.txt","w") rec=fh.read(); for a in rec:     if (a.isdigit() != True):         print(a,end=' ')         fw.write(a) fh.close() fw.close()</pre>	3														

28	<p>(a) SELECT NAME FROM TEACHER WHERE NAME LIKE “__0%”; _ and % are the wildcards for pattern matching.</p> <p>(b) i.</p> <table border="1" style="margin-left: 40px;"> <tr><td style="width: 20px; text-align: center;">3</td><td>DELHI</td></tr> <tr><td style="text-align: center;">2</td><td>MUMBAI</td></tr> <tr><td style="text-align: center;">1</td><td>MADRAS</td></tr> </table> <p>ii. 50000,70000 iii.11 iv.</p> <table border="1" style="margin-left: 40px;"> <tr><td style="width: 20px; text-align: center;">MOBILE</td><td>MUMBAI</td><td style="text-align: center;">70000</td></tr> <tr><td style="text-align: center;">MOBILE</td><td>MUMBAI</td><td style="text-align: center;">25000</td></tr> </table>	3	DELHI	2	MUMBAI	1	MADRAS	MOBILE	MUMBAI	70000	MOBILE	MUMBAI	25000	1+2
3	DELHI													
2	MUMBAI													
1	MADRAS													
MOBILE	MUMBAI	70000												
MOBILE	MUMBAI	25000												

29	<pre>def EVEN_LIST(L):     evenList=[]     for i in L:         if i%2==0:             evenList.append(i)     return(evenList)</pre>	3
----	---	---

30	<pre>def PUSH(S):     for i in L:         if i%2!=0:             S.append(i)     return(S)  def POP():     num=len(S)     while len(S)!=0:         dele=S.pop()         print(dele)         num=num-1     else:         print("empty")  def Push(Bitem):     for i,j in Bitem.items():         if j&lt;50:             S.append(i)</pre> <p style="text-align: center;">OR</p>	3
----	--	---

<b>SECTION-D</b>	
------------------	--

31	<p>(i)admin; it contains the max number of systems. to reduce traffic (ii)</p> <div style="border: 1px solid black; padding: 10px; margin: 10px 0;"> <pre> graph LR     ADMIN[ADMIN] --- ENGINEERING[ENGINEERING]     ADMIN --- BUSINESS[BUSINESS]     ADMIN --- MEDIA[MEDIA] </pre> </div> <p>(iii)firewall (iv) (c) Video Conferencing (v) POP and SMTP</p>	5
----	---	---

32	<p>(a) ['and y', 'u will!'] (b)</p> <pre>import mysql.connector as pymysql #statement 1 dbcon=pymysql.connect(host="localhost",user="root",passwd="sia@1928", database=' School' ) # statement 2 if dbcon.isconnected()==False:     print("Error in establishing connection:") cur=dbcon.cursor() # statement 3</pre>	5
----	---	---

	<pre> query="select * from stmaster" cur.execute(query) # statement 4 resultset=cur.fetchmany(3) for row in resultset:     print(row) dbcon.close() # statement 5  OR  (a)hELLl&amp;EeEeryoE (b) Statement 1: con1.cursor() Statement 2: mycursor.execute("select * from student where Marks&gt;75") Statement 3: mycursor.fetchall() </pre>	
33	<p>Advantage of a csv file:</p> <ul style="list-style-type: none"> <li>• It is human readable – can be opened in Excel and Notepad applications</li> <li>• It is just like text file</li> </ul> <p>Program:</p> <pre> import csv def ADD():     fout=open("record.csv","a",newline="\n")     wr=csv.writer(fout)     empid=int(input("Enter Employee id :: "))     name=input("Enter name :: ")     mobile=int(input("Enter mobile number :: "))     lst=[empid,name,mobile] -----1/2 mark     wr.writerow(lst) -----1/2 mark     fout.close() def COUNTR():     fin=open("record.csv","r",newline="\n")     data=csv.reader(fin)     d=list(data)     print(len(d))     fin.close() ADD() COUNTR()  OR  Binary file: <ul style="list-style-type: none"> <li>• Extension is .dat</li> <li>• Not human readable</li> <li>• Stores data in the form of 0s and 1s</li> </ul> <p>CSV file</p> <ul style="list-style-type: none"> <li>• Extension is .csv</li> <li>• Human readable</li> <li>• Stores data like a text file</li> </ul> <p>Program:</p> <pre> import csv def add():     fout=open("furdata.csv","a",newline='\n')     wr=csv.writer(fout)     fid=int(input("Enter Furniture Id :: "))     fname=input("Enter Furniture name :: ")     fprice=int(input("Enter price :: "))     FD=[fid,fname,fprice]     wr.writerow(FD)     fout.close() def search():     fin=open("furdata.csv","r",newline='\n')     data=csv.reader(fin)     found=False </pre> </pre>	5

	<pre> print("The Details are") for i in data:     if int(i[2])&gt;10000:         found=True         print(i[0],i[1],i[2]) if found==False:     print("Record not found") fin.close() add() print("Now displaying") search() </pre>	
	<b>SECTION-E(4 marks)</b>	0
34	<pre> import csv f=open("pl.csv","w") cw=csv.writer(f) ch="Y" while ch=="Y":     l=[]     pi=int(input("enter dvd id "))     pnm=input("enter dvd name ")     sp=int(input("enter qty "))     p=int(input("enter price(in rupees) "))     l.append(pi)     l.append(pnm)     l.append(sp)     l.append(p)     cw.writerow(l)     ch=input("do you want to enter more rec(Y/N): ").upper()     if ch=="Y":         continue     else:         break f.close() f=open("pl.csv","r+") cw=list(csv.reader(f)) for i in cw:     if l[3]&gt;25:         print(i) f.close() </pre>	4
35	<ol style="list-style-type: none"> <li>1. SELECT TEACHERNAME, PERIODS FROM SCHOOL WHERE PERIODS&gt;25;</li> <li>2. SELECT * FROM SCHOOL;</li> <li>3. SELECT DISTINCT DESIGNATION FROM ADMIN;</li> <li>4. SELECT TEACHERNAME.CODE DESIGNATION FROM SCHOOL.CODE = ADMIN.CODE WHERE GENDER = MALE;</li> </ol>	4