

KENDRIYA VIDYALAYA SANGATHAN, PATNA REGION
PRE-BOARD I (2023 - 24)

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	
	SECTION B (2 Marks)		
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	BUS TOPOLOGY	STAR TOPOLOGY	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.	
	OR		

	DOMAIN OF COMPARISON	CIRCUIT SWITCHING	PACKET SWITCHING	
	Definition	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.	
	Data processing	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.	
	Flexibility	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.	
	Utility	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	CHAR	VARCHAR		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="204 123 762 163">Where Clause in SQL</th> <th data-bbox="762 123 1348 163">Having Clause in SQL</th> </tr> </thead> <tbody> <tr> <td data-bbox="204 163 762 232">Filter table based data catering to specific condition</td> <td data-bbox="762 163 1348 232">Group based data under set condition</td> </tr> <tr> <td data-bbox="204 232 762 271">Applicable without GROUP BY clause</td> <td data-bbox="762 232 1348 271">Does not function without GROUP BY clause</td> </tr> <tr> <td data-bbox="204 271 762 309">Row functions</td> <td data-bbox="762 271 1348 309">Column functions</td> </tr> <tr> <td data-bbox="204 309 762 347">Select, update and delete statements</td> <td data-bbox="762 309 1348 347">Only select statement</td> </tr> <tr> <td data-bbox="204 347 762 385">Applied before GROUP BY clause</td> <td data-bbox="762 347 1348 385">Used after GROUP BY clause</td> </tr> <tr> <td data-bbox="204 385 762 454">Used with single row operations such as Upper, Lower and so on</td> <td data-bbox="762 385 1348 454">Applicable with multiple row functions such as Sum, count and so on</td> </tr> </tbody> </table> <p data-bbox="692 445 735 474" style="text-align: center;">OR</p> <p data-bbox="204 479 1410 544">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																
SECTION C (3 Marks)																	
26	<p data-bbox="252 665 1129 696">(a) ALTER TABLE PRODUCT ADD TOTAL PRICE NUMBER (10,2).</p> <p data-bbox="252 701 295 732">(b)</p> <p data-bbox="328 730 352 757">(i)</p> <table border="1" data-bbox="368 761 616 792"> <tr> <td data-bbox="368 761 552 792">VICE PRINCIPAL</td> <td data-bbox="552 761 616 792">01</td> </tr> </table> <p data-bbox="328 801 352 828">(ii)</p> <table border="1" data-bbox="368 833 411 864"> <tr> <td data-bbox="368 833 411 864">16</td> </tr> </table> <p data-bbox="328 873 352 900">(iii)</p> <table border="1" data-bbox="368 904 533 969"> <tr> <td data-bbox="368 904 533 936">UMESH</td> </tr> <tr> <td data-bbox="368 936 533 969">YASH RAJ</td> </tr> </table> <p data-bbox="328 978 352 1005">(iv)</p> <table border="1" data-bbox="368 1010 533 1075"> <tr> <td data-bbox="368 1010 533 1041">5 MALE</td> </tr> <tr> <td data-bbox="368 1041 533 1075">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 1081 239 1113">(a)</p> <pre data-bbox="204 1115 1157 1406">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="647 1413 691 1444" style="text-align: center;">OR</p> <p data-bbox="204 1449 239 1480">(b)</p> <pre data-bbox="204 1482 786 1774">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 (a) SELECT NAME FROM TEACHER WHERE NAME LIKE “__0%”;
 _ and % are the wildcards for pattern matching.
 (b) i.

3	DELHI
2	MUMBAI
1	MADRAS

ii. 50000,70000
 iii.11
 iv.

MOBILE	MUMBAI	70000
MOBILE	MUMBAI	25000

29 `def EVEN_LIST(L):`
 `evenList=[]`
 `for i in L:`
 `if i%2==0:`
 `evenList.append(i)`
 `return (evenList)`

30 `def PUSH(S):`
 `for i in L:`
 `if i%2!=0:`
 `S.append(i)`
 `return (S)`

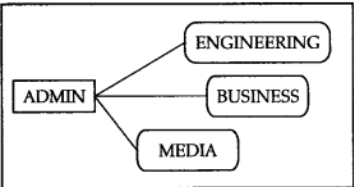
`def POP():`
 `num=len(S)`
 `while len(S)!=0:`
 `dele=S.pop()`
 `print(dele)`
 `num=num-1`
 `else:`
 `print("empty")`

OR

`def Push(Bitem):`
 `for i,j in Bitem.items():`
 `if j<50:`
 `S.append(i)`

SECTION-D

31 (i)admin; it contains the max number of systems. to reduce traffic
 (ii)



```

    graph LR
      ADMIN[ADMIN] --- ENGINEERING[ENGINEERING]
      ADMIN --- BUSINESS[BUSINESS]
      ADMIN --- MEDIA[MEDIA]
    
```

(iii) firewall
 (iv) (c) Video Conferencing
 (v) POP and SMTP

32 (a) [`'and y'`, `'u will!'`]
 (b)

```

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> 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&EeEeryoE (b) Statement 1: con1.cursor() Statement 2: mycursor.execute("select * from student where Marks>75") Statement 3: mycursor.fetchall() </pre>	
33	<p>Advantage of a csv file:</p> <ul style="list-style-type: none"> • It is human readable – can be opened in Excel and Notepad applications • It is just like text file <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() </pre> <p style="text-align: center;">OR</p> <p>Binary file:</p> <ul style="list-style-type: none"> • Extension is .dat • Not human readable • Stores data in the form of 0s and 1s <p>CSV file</p> <ul style="list-style-type: none"> • Extension is .csv • Human readable • Stores data like a text file <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>	5

	<pre> print("The Details are") for i in data: if int(i[2])>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>	
	SECTION-E(4 marks)	0
<p>34</p>	<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]>25: print(i) f.close() </pre>	<p>4</p>
<p>35</p>	<ol style="list-style-type: none"> 1. SELECT TEACHERNAME, PERIODS FROM SCHOOL WHERE PERIODS>25; 2. SELECT * FROM SCHOOL; 3. SELECT DISTINCT DESIGNATION FROM ADMIN; 4. SELECT TEACHERNAME.CODE DESIGNATION FROM SCHOOL.CODE = ADMIN.CODE WHERE GENDER = MALE; 	<p>4</p>