

**KENDRIYA VIDYALAYA SANGATHAN, AGRA REGION**

Class: XII Session: 2023-24

Computer Science (083)

Second Pre Board Exam 2023 (Theory)

Time allowed: 3 Hours

Maximum Marks: 70

Ques No	Question and Answers	Total Marks
	<b>SECTION A</b>	<b>1</b>
1	<b>d)print()</b>	<b>1</b>
2	<b>True</b>	<b>1</b>
3	<b>c)23.0</b>	<b>1</b>
4	<b>d) None of the above</b>	<b>1</b>
5	c) ALTER table emp ADD bonus int;	<b>1</b>
6	(a)32 bits	<b>1</b>
7	d) error	<b>1</b>
8	b) List	<b>1</b>
9	b)(['salary', 'dept', 'age', 'name'])	<b>1</b>
10	<b>a) 3,4</b>	<b>1</b>
11	B) Modem	<b>1</b>
12	<b>a)continue</b>	<b>1</b>
13	<b>a) OverflowError</b>	<b>1</b>
14	A. Fixed, Variable	<b>1</b>
15	<b>a)SMTP</b>	<b>1</b>
16	<b>a) fp.seek(offset, 0)</b>	<b>1</b>
17	(a) Both A and R are true and R is the correct explanation for A	<b>1</b>
18	A. A is True but R is False	<b>1</b>
	<b>SECTION B</b>	
19	<b>(a) TCP-Transmission Control Protocol/ Internet Protocol VoIP-Voice over Internet Protocol</b> <b>(b) FTP is used for sharing files over internet.</b> <b>OR</b> <b>(a) One mark for any one correct difference.</b> <b>(b) One mark for any one correct difference.</b>	<b>2</b>
20	<b>def sum(arg1,arg2):</b> <b>total=arg1+arg2 ;_ # No need of semicolon at two places</b> <b>print("Total:",total)</b> <b>__ return total ;__ # wrong indentation for return statement</b> <b><u>a</u>=sum(10,20) #Value returned from function is not received</b>	<b>2</b>

	(½ mark for each correct correction made and underlined.)	
21	['E', 'a', 'i', 'a', 'i', 'n']	2
22	<pre>def change(L):     EL=[]     OL=[]     for l in range(0,len(L)):         if L[i]%2==0:             EL.append(L[i])         else:             OL.append(L[i])     print("List of even elements:",EL)     print ("List of odd elements:",OL)</pre> <p><i>Note: Any other relevant and correct code may be marked</i></p> <p style="text-align: center;"><b>OR</b></p> <pre>def INDEX_LIST(S):     indexList=[]     for i in range(len(S)):         if S[i] in 'aeiouAEIOU':             indexList.append(i)     return indexList</pre> <p><i>Note: Any other relevant and correct code may be marked</i></p>	2
23	<pre>(i) L1.insert(1,150) (ii) message.upper( )</pre> <p style="text-align: center;"><b>OR</b></p> <pre>import math print(math.factorial(5))</pre>	2
24	<p><b>SQL query to create the table HRDATA:</b> CREATE TABLE HRDATA (Ecode int, Ename char(50),Desig char(5),remn int);</p> <p><b>SQL query to insert given data in to the table HRDATA:</b> INSERT INTO HRDATA VALUES(80008,"Arjun","Admin",55000);</p> <p style="text-align: center;"><b>OR</b></p> <p><b>SQL query to remove the column Quantity from table CHStore:</b> ALTER TABLE CHStore drop column Quantity;</p> <p><b>SQL query to display the structure of the table CHStore:</b> DESC CHStore; or DESCRIBE CHStore;</p>	2
25	<pre>22 # 40 # 9 # 13 #</pre> <p>(½ mark for the correct digit with a #)</p>	2
<b>SECTION C</b>		
26	<b>New String is: iNdIA%****</b>	3
27	<pre>a)475 b)6</pre>	3

	<p><b>c)68</b>  <i>1 mark for each correct answer.</i></p>	
28	<pre>def count_W_H():     f = open ("Country.txt", "r")     W,H = 0,0     r = f.readlines()     for x in r:         if x[0] == "W" or x[0] == "w":             W=W+1         elif x[0] == "H" or x[0] == "h":             H=H+1      f.close()     print ("W or w :", W)     print ("H or h :", H)</pre> <p style="text-align: center;"><b>OR</b></p> <pre>def countwords():     s = open("Quotes.txt","r")     f = s.read()     z = f.split ()     count = 0     for i in z:         count = count + 1     print ("Total number of words:", count)</pre> <p>Note: Using of any correct code giving the same result is also accepted.</p>	3
29	<p>a) <b>UPDATE EMPLOYEE</b>  <b>SET SALARY=SALARY+(SALARY*.05)</b>  <b>WHERE JOB TYPE="Clerk";</b></p> <p>b) <b>SELECT * FROM EMPLOYEE WHERE DATEOFJOIN</b>  <b>BETWEEN '1985-01-01' AND '2000-03-31';</b></p> <p>c) <b>DELETE FROM EMPLOYEE WHERE JOB TYPE='Manager';</b></p>	3
30	<pre>status=[] def Push_element(cust):     if cust[2]=="Goa":         L1=[cust[0],cust[1]]         status.append(L1)  def Pop_element ():     num=len(status)     while len(status)!=0:         dele=status.pop()         print(dele)         num=num-1      else:         print("Stack Empty")</pre> <p><b>(1.5 marks for correct Push_element() and 1.5 marks for correct Pop_element())</b></p>	3

SECTION D		
31	a) SELECT COUNT(*) FROM EMP GROUP BY DEPTNO; b) SELECT JOB, COUNT(*) FROM EMP WHERE COUNT(*)<3 GROUP BY JOB; c) SELECT ENAME, DNAME FROM EMP,DEPT WHERE EMP.DEPTNO=DEPT.DEPTNO; d) SELECT MAX(SAL) FROM EMP WHERE JOB='MANAGER';	4
32	<pre> import csv def ADDPROD():     fout=open("product.csv","a",newline="\n")     wr=csv.writer(fout)     prodid=int(input("Enter Product id :: "))     name=input("Enter Product name : ")     price=int(input("Enter Product Price : "))     lst=[ prodid, name, price]     wr.writerow(lst)     fout.close() def COUNTPROD():     fin=open("product.csv", "r",newline="\n")     data=csv.reader(fin)     d=list(data)     print(len(d))     fin.close() ADD() COUNTR() </pre> <p>(½ mark for importing csv module  1 ½ marks each for correct definition of ADD() and COUNTR()  ½ mark for function call statements)</p>	4
SECTION E		
33	<p>(a) Suggest a cable layout of connection between the blocks.</p> <pre> graph TD     Business --- TechnologyBlock[Technology Block]     LawBlock --- Business     HRCentre[HR Centre] --- TechnologyBlock </pre> <p>(b) Ans : HR centre because it consists of the maximum number of computers to house the server.  (c) Ans: Switch/ Hub should be placed in each of these blocks.  (d) Ans : MAN  (e) Ans : Bus</p>	5
34	(i) Binary file:	5

- Extension is .dat
- Not human readable
- Stores data in the form of 0s and 1s

#### CSV file

- Extension is .csv
- Human readable
- Stores data like a text file

```
(ii) import pickle
      def AddStudents():
          F= open("STUDENT.DAT",'wb')
          while True:
              Rno = int(input("Rno :"))
              Name = input("Name : ")
              Percent = float(input("Percent :"))
              L = [Rno, Name, Percent]
              pickle.dump(L,F)
              Choice = input("enter more (y/n): ")
              if Choice in "nN":
                  break

          F.close()
      def GetStudents():
          Total=0
          Countrec=0
          Countabove75=0
          with open("STUDENT.DAT","rb") as F:
              while True:
                  try:
                      R = pickle.load(F)
                      Countrec+=1
                      Total+=R[2]
                      if R[2] > 75:
                          print(R[1], " has percent =",R[2])
                          Countabove75+=1
                  except:
                      break
              if Countabove75==0:
                  print("No student has percentage more
than 75")
          print("average percent of class = ", Total /
Countabove75)
      AddStudents()
      GetStudents()
```

**OR**

- (a) Advantage: Faster and no conversion required  
Disadvantage: Can't read by users directly

	<pre> (b) 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 </pre>	
<p>35</p>	<p>(i) For each attribute of a relation, there is a set of permitted values, called Domain of that attribute. Example: any valid example</p> <p>(ii) <b>import mysql.connector as mysql</b>  <b>def sql_data():</b>  <b>con1=mysql.connect(host="localhost",user="root",</b>  <b>password="root@123", database="management")</b>  <b>mycursor= con1.cursor()</b>  <b>print("Employees with salary greater than 53500 are : ")</b>  mycursor.execute("select * from <b>Employees</b> where  EmpSalary &gt;53500")  <b>data = mycursor.fetchall()</b>  <b>for i in data:</b>  <b>print(i)</b>  <b>print()</b></p> <p style="text-align: center;"><b>OR</b></p> <p>(i) <b>Degree is the no of columns in a relation while</b>  <b>Cardinality is the no of tuples in a relation.</b>  <b>Any example</b></p> <p>(ii) <b>import mysql.connector as mydb</b>  mycon = mydb.connect (host = "localhost", user = "root",  passwd = "system", database = "Admin")  cursor = mycon.cursor ( )</p>	<p>5</p>

	<pre>sql = "UPDATE TRAINER SET SALARY = SALARY + 2000 WHERE TNAME = 'SUNAINA'" cursor. execute(sql) mycon.commit ( ) mycon.close( )</pre>	
--	---	--