

**SET-2**  
**KENDRIYA VIDYALAYA SANGATHAN (LUCKNOW REGION)**  
**XII- PRE-BOARD EXAMINATION 2023-24**  
**Computer Science (083)**

*Time allowed: 3 hours*

*Maximum marks: 70*

**Note – Please award full marks for any alternative correct code / set of instruction especially in Python programming.**

**MARKING SCHEME**

Section – A		
Q01.	Ans: True	(1)
Q02.	Ans: (c) _123	(1)
Q03.	Ans: (b) (1, 2, [1, 3.5], 3)	(1)
Q04.	Ans: (b) False	(1)
Q05.	Ans: (d) ['Economy', 'Digital']	(1)
Q06.	Ans: (d) None of these	(1)
Q07.	Ans: (c) ALTER	(1)
Q08.	Ans: (c) DROP	(1)
Q09.	Ans: (a) Statement1	(1)
Q10.	Ans: (c) Cardinality	(1)
Q11.	Ans: (a) When you open a file for reading, if the file does not exist, an error occurs.	(1)
Q12.	Ans: (d) LIKE	(1)
Q13.	Ans: (c) URL	(1)
Q14.	Ans: (a) True	(1)
Q15.	Ans: (d) SELECT MAX(SALARY) AND MEAN(SALARY) FROM EMPLOYEE;	(1)
Q16.	Ans: (b) mycon.connect(host="localhost",user="root",passwd="MyPass",database="test")	(1)
Q17.	Ans: (b) Both A and R are true and R is not the correct explanation for A	(1)
Q18.	Ans: (c) A is True but R is False	(1)

**Section – B**

Q19. **Ans:**

(2)

```
def factorial(N):  
    fact=1  
    while N>0:  
        fact*=N  
        N=N-1  
    print("Factorial=",fact)  
  
#main()  
  
factorial(5)
```

Q20. Differentiate between Star Topology and Bus Topology. Write two points of difference.

(2)

**Ans:**

S.No.	Star topology	Bus Topology
1	In Star topology all the devices are connected to a central hub/node.	In Bus topology each device in the network is connected to a single cable which is known as the backbone
2	In Star topology if the central hub fails then the whole network fails.	In Bus topology the failure of the network cable will cause the whole network to fail.

OR

Write two point of difference between Bridge and Gateway.

**Ans:**

S.No.	Bridge	Gateway
1	Bridge connect two different Networks working on the same protocol	Gateway is used to connect two different Networks working on the different protocols
2	In bridge, format of packet is not changed	In Gateway format of packet is changed.

- Q21. **Ans:** Output will be as follows (1)
- (a) oetcnlg hl rwn p (1)
- (b) **{'name': 'aditya', 'age': 16}**  
**{'name': 'aditya', 'age': 16, 'address': 'Mumbai'}**
- Q22. Degree: The number of attributes or columns in a relation is called the Degree of the relation. (2)
- Attribute: An attribute is a specification that defines a property of an object (column / fields of a relation) Example: Any valid example to show degree and attribute.
- Q23. (a) (i) File Transfer Protocol (ii) Post Office Protocol (1)
- (b) A Media Access Control address (MAC address) is a hardware identifier that uniquely identifies each device on a network. Primarily, the manufacturer assigns it. They are often found on a device's network interface controller (NIC) card. (1)
- Q24. **Ans: [2, 26, 23, 18]** (1/2 mark for each correct digit) OR (2)
- Ans: (4, 6, 8, 7, 5, 3)**(1/2 marks for each two consecutive digits & ½ marks for parenthesis)
- Q25. (1 MARKS FOR DIFFERENCE AND 1 MARKS FOR EXAMPLE) (2)

S.No.	WHERE	HAVING
1	It is used to fetch the records from the table based on the specified condition.	It is used to fetch the record from the groups based on the specified condition.
2	It is used before group by or order by	It is used along with group by clause
3	Any valid appropriate example	Any valid appropriate example

OR

The GROUP BY clause is used to get the summary data based on one or more group of records. The groups can be formed on one or more columns. For example, the GROUP BY query will be used to count the number of employees in each department or to get the department wise total salaries etc. It uses HAVING clause to retrieve records based on specified condition for the given column(s) of the group by clause.

Example: any valid example

### Section – C

BNO	NAME	CITY	ITEMCODE	INAME	PRICE
1001	Ashok	Delhi	11	HDD	2500
1001	Ashok	Delhi	12	Keyboard	850
1001	Ashok	Delhi	13	RAM	1100
1002	Manish	Jaipur	11	HDD	2500
1002	Manish	Jaipur	12	Keyboard	850
1002	Manish	Jaipur	13	RAM	1100
1003	Rahul	Lucknow	11	HDD	2500
1003	Rahul	Lucknow	12	Keyboard	850
1003	Rahul	Lucknow	13	RAM	1100

(a) 1 mark for correct result.

(b)

(2)

(1/2 mark for each correct output)

(i) Manager

Director

Clerk

(ii) Delhi 90000

Mumbai 175000

Kolkata 85000

(iii) Peter 45000

Jack 85000

Harry 90000

(iv) Harry 90000 Delhi

Q27. (a)

def COUNT\_W5():

```

obj = open("story1.txt","r")

text = obj.read()

L = text.split()

count = 0

print("Six characters words are: - ", end = "")

for i in L:

    if len(i) == 6:

        print(i , end=" ")

        count += 1

print("\n\nThe total no of words with length of 6 characters is: ",count)

```

COUNT\_W5()

Q27. OR (b)

(3)

```

def VC_COUNT():

    obj=open("THEORY.TXT","r")

    text=obj.read()

    VCOUNT=0

    CCOUNT=0

    for i in text:

        if( i.isalpha()):

            if i in ('a','e','i','o','u') or i in ('A','E','I','O','U'):

                VCOUNT+=1

            else:

                CCOUNT+=1

    print("Total vowels are: ",VCOUNT)

    print("Total consonants are: ",CCOUNT)

VC_COUNT()

```

Q28. (a) (1/2 marks for each correct output)

(3)

(i) CODING 2

NETWORKING 1

ANALYSIS 1

TESTING 1

(ii) '2017-04-10' '2003-05-12'

(iii) RAVI CODING 45000

SUMAN NETWORKING 50000

MONU CODING 57800

RUPALI ANALYSIS 39600

ROHIT TESTING 33700

(iv) SUMAN 2500

REPALI 3000

½ mark for each output

(b) mysql> DESC EMPLOYEE; (1 mark for correct command)

Q29. def INTERCHANGE(L):

(3)

```
    ChangedList=L
```

```
    i=0
```

```
    if len(ChangedList)%2==0:
```

```
        while i<=(len(ChangedList)-1):
```

```
            ChangedList[i],ChangedList[i+1]=ChangedList[i+1],ChangedList[i]
```

```
            i=i+2
```

```
    else:
```

```
        while i<=(len(ChangedList)-2):
```

```
            ChangedList[i],ChangedList[i+1]=ChangedList[i+1],ChangedList[i]
```

```
            i=i+2
```

```
return ChangedList
```

```
List=[5,6,7,8,9]
```

```
print(List)
```

```
NewList = INTERCHANGE(List)
```

```
print(NewList)
```

Q30. def Push\_record(): # (1½ mark for correct push element)

(3)

```
for i in List:
```

```
    if i[2]=="Delhi":
```

```
Record.append(i)
```

```
print(Record)
```

```
def Pop_record(): # (1½ mark for correct push element)
```

```
while True:
```

```
    if len(Record)==0:
```

```
        print('Empty Stack')
```

```
        break
```

```
    else:
```

```
        print(Record.pop())
```

OR

```
data = [1,2,3,4,5,6,7,8]
```

```
stack = []
```

```
def push(stack, data):
```

```
    for x in data:
```

```
        if x % 2 == 0:
```

```
            stack.append(x)
```

```
def pop(stack):
```

```
    if len(stack)==0:
```

```
        return "stack empty"
```

```
    else:
```

```
        return stack.pop()
```

```
push(stack, data)
```

```
print(pop(stack))
```

½ mark should be deducted for all incorrect syntax. Full marks to be awarded for any other logic that produces the correct result.

Section – D

Q31. (i) MCODE unique values (1 mark) (4)

(ii) Degree = 4 (after removing one column) (1/2 mark)

Cardinality = 7 (after 2 more record added) (1/2 mark)

(iii) (a) `mysql>alter table MOBILES add GST int;` (1 mark)

(b) `mysql>update MOBILES set GST=(PRICE*0.18)` (1 mark)

OR

(iii) (a) `mysql>insert into MOBILES values('M06','iPHONE13','APPLE',110000,'2022-03-01');`

(1 mark)

(b) `mysql>delete from MOBILES where MODEL='NARZO50';` (1 mark)

Q32 (i) pickle (1 mark for correct module) (4)

(ii) Statement2: `f=open("student.dat","rb")` (1 mark)

(iii) Statement3: `rec=pickle.load(f)` (1 mark)

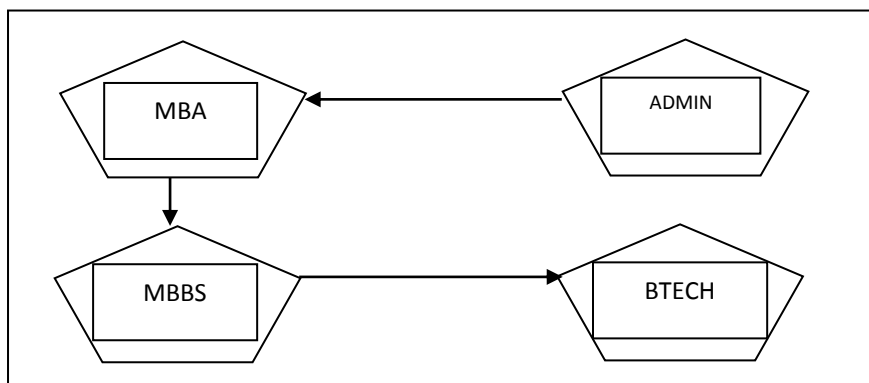
Statement4: `if(rec[0]==7):` (1 mark)

Section - E

Q33. (i) Admin block, Justification – maximum number of computers (5)

(ii) Firewall

(iii)



(iv) (a) Switch/Hub in each block to connect all the computers of that block.



(b) Repeater between MBBS and BTECH block due to 190 meter of distance apart from each other..

(v) VoIP protocol

Q34. (a) Ans: Comp.Sc.- 083#083 (1 mark for comp.sc. and 1 marks for rest) (5)

(b) Statement1: mycursor=con1.cursor()

Statement2: mycursor.execute(query)

Statement3: con1.commit()

(1 mark for each correct statement)

OR

(a) Ans: 9\$8\$25\$27\$ (1 mark for first 5 characters and 1 mark for next 5 characters)

(b) Statement1: mycursor=con1.cursor()

Statement2: mycursor.execute("select \* from Product where Price>500")

Statement3: data=mycursor.fetchall()

(1 mark for each correct statement)

Q35. **CSV and TXT files store information in plain text.** The first row in the file defines the names (5)  
for all subsequent fields. In CSV files, fields are always separated by commas. In TXT files, fields can be separated with a comma, semicolon, or tab. **(1 marks)**

```
import csv #( 1/2 mark for import csv)
```

```
def INSERT(): #(1 1/2 for correct definition of function)
```

```
    fh=open("Sdetails.csv",'w')
```

```
    swriter=csv.writer(fh)
```

```
    ans='y'
```

```
    swriter.writerow(['RollNo','Name','Class','Marks'])
```

```
    ans='y'
```

```
    while ans=='y' or ans=='Y':
```

```
        rn=int(input("Enter Roll No"))
```

```
        nm=input("Enter Name")
```

```
        cl=int(input("Enter class"))
```

```
        mk=int(input("Enter marks"))
```

```

srec=[rn,nm,cl,mk]
swriter.writerow(srec)
ans=input("if you want to enter more record press Y/N")
fh.close()

```

def COUNTROW(): #**(1 ½ for correct definition of function)**

```

with open("Sdetails.csv",'r', newline='\r\n') as fh:

```

```

    sreader=csv.reader(fh)

```

```

    count=0

```

```

    for r in sreader:

```

```

        count+=1

```

```

    print("total record = ",count)

```

```

fh.close()

```

#main program

INSERT()

COUNTROW() ( ½ mark for both function call statements)

OR

Syntax to create reader object:

Object=csv.reader("filehandle") ( ½ mark)

Object=csv.writer("filehandle")( ½ mark)

import csv #( ½ mark for import csv)

def ADD( ): # **(1½ mark for correct definition)**

```

f=open("empdata.csv",'w')

```

```

swriter=csv.writer(f)

```

```

ans='y'

```

```

swriter.writerow(['Eid','EName','Salary','City'])

```

```

ans='y'

```

```

while ans=='y' or ans=='Y':

```

```

    eid=int(input("Enter employee id"))

```

```

    enm=input("Enter Employee Name")

```

```

    es=int(input("Enter Salary"))

```

```
ec=input("Enter City")
erec=[eid,enm,es,ec]
writer.writerow(erec)
ans=input("if you want to enter more record press Y/N")
f.close()
```

```
def SEARCH(): #(1 ½ mark for correct definition)
```

```
with open("empdata.csv",'r', newline="\r\n") as fh:
```

```
ereader=csv.reader(fh)
```

```
for r in ereader:
```

```
    if int(r[2])>50000:
```

```
        print(r[0],r[1],r[2],r[3])
```

```
        break
```

```
else:
```

```
    print("Employee Data not found")
```

```
#main (1/2 marks of functions calling)
```

```
ADD()
```

```
SEARCH()
```

**\*\*\*\*\*End Of Paper\*\*\*\*\***