KENDRIYA VIDYALAYA SANGATHAN REGIONAL OFFICE LUCKNOW

PRE-BOARD EXAMINATION 2025-26

CLASS: XII **SUBJECT: COMPUTER SCIENCE**

TIME: 3 HOURS **M. MARKS: 70**

MODEL ANSWER KEY

Specific Instructions:

- The answers given in the marking scheme are SUGGESTIVE. Examiners are requested to award marks for all alternative correct Solutions/Answers conveying a similar meaning
- All programming questions have to be answered with respect to Python only
- In Python, ignore case sensitivity for identifiers (Variable / Functions / Structures / Class Names)
- In Python indentation is mandatory, however, the number of spaces used for indenting may vary
- In SQL related questions – both ways of text/character entries should be acceptable for Example: "AMAR" and 'amar' both are acceptable.
- In SQL related questions – all date entries should be acceptable for Example: 'YYYY-MM-DD', 'YY-MM-DD', 'DD-Mon-YY', "DD/MM/YY", 'DD/MM/YY', "MM/DD/YY", 'MM/DD/YY' and {MM/DD/YY} are correct.
- In SQL related questions – semicolon should be ignored for terminating the SQL statements
- In SQL related questions – ignore case sensitivity
- In SQL output questions – ignore the column headers
- In SQL output questions – ignore the order of rows until ORDER BY is specified

Q No	Section-A (21x1 = 21 marks) *(1 Mark for writing the correct option)*	Marks
1	a) True	1
2	a) True	1
3	b) 4.6	1
4	c) ['Chi','ese Co','ti','e','tal']	1
5	b) False#True	1
6	Both b) and c)	1
7	b) ('w', 'or', 'k is worship')	1
8	2 rl	1
9	c) 3	1
10	c) Hilcome	1
11	True	1

12	c) 12#15%	1
13	b) Alter table	1
14	b) INSERT	1
15	a) PRIMARY, CANDIDATE	1
16	d) both a and c	1
17	d) Telnet	1
18	d) Gateway	1
19	False	1

Q20 and Q21 are Assertion(A) and Reason(R) based questions.
Mark the correct choice as:
a) Both A and R are True and R is the correct explanation for A.
b) Both A and R are True and R is not the correct explanation for A.
c) A is True but R is False.
d) A is False but R is True.

20	c) A is True but R is False.	1
21	c) A is True but R is False.	1

Section - B (7x2 = 14 marks)

22 A mutable object allow in-place change in the value whereas immutable object doesn't allow in-place change in the value. 1
Mutable object : [1,2], {1:1,2:2}
Immutable object : (1,2), '123' 1

(1 Mark for writing the correct explanation) OR
(1 Mark for writing the correct example)
OR
(Full 2 Marks for correct explanation with the help of a suitable example) 1

OR

The main difference is that break **exits the entire loop** while continue **skips the current iteration** and proceeds to the next one.
break is used to stop the loop completely, whereas continue is used to skip over certain iterations based on a condition, allowing the loop to keep running. 1

curr = 0
# Loop till curr < 10 	
while curr < 10:
  print(curr)
  curr += 1
  # If curr = 5, break out of the loop
  if curr == 5:
  break
print("Thanks")
output:
	# Loop till curr <10
	for curr in range(10):
	# If curr = 5, then continue to the next iteration
	  if curr== 5:
	  continue
	  print(curr)
	print("Thanks")
	output:
	0

0 1 2 3 4 Thanks **Here loop terminated for all the next iterations after the curr=5**	1 2 3 4 6 7 8 9 Thanks **Here current iteration of loop terminated for the curr=5 and continues next iterations.**

(1 Mark for writing the correct difference)
(1 Mark for writing the correct example)
OR
(Full 2 Marks for correct difference with the help of a suitable example)

23 **def Calc(a):** 2
s=0,m=0,n=0
for i in range (0,a):
if i%**2 = = 0**:
s=s+1
elif i%5= = **0 :**
m=m+2
else:
n=n+i
print(s,m,n)
Calc(15)

(Full 2 Marks for writing all required corrections)
OR
(1 Mark for only identification of all/any four errors without correction)

24 a. L1.extend(L2) 1+1
b. L1.pop()

OR

a) del D['Mumbai']
b) print(S.split())

(1 Mark for writing each correct statement)

25 ['Ashish', 'Jaya'] 2

OR

hARRYpOTTER$@$$

(2 Marks for writing the correct output)

26 def countNow(PLACES): 2

```
    for j in PLACES.values():
        if len(j)>5:
            print(j.upper())
PLACES={1: "Delhi", 2: "London", 3: "Paris" ,4: "New York", 5: "Doha" }
countNow(PLACES)
```

(½ Mark for correct function header)
(½ Mark for correct loop)
(½ Mark for correct logic of extracting the more than 5 characters long words)
(½ Mark for print in uppercase)

27 (i) Foreign key depends on the primary value of another table. 2
(ii) There are NULL values in the HOBBY column.

OR

(i) Distinct keyword is used to remove duplicate values from selecting the record from the table.
(ii) ALTER TABLE MOBILE ADD PRIMARY KEY(M_ID);.

(1 Mark for each correct answer/ statement)

28 **i) Star Topology :** 2

Advantages

- Simplicity: Easy to set up and understand.
- Easy Troubleshooting: If one node fails, it doesn't affect the entire network, making it easier to troubleshoot.
- Scalability: Adding or removing nodes is straightforward without disrupting the network.

Disadvantages

- Single Point of Failure: If the central hub fails, the entire network goes down.
- Cost: Requires more cable than bus topology, leading to higher costs in larger networks.
- Limited Scalability: The scalability of the network depends on the capacity of the central hub.

ii) Tree Topology

Advantages

- Scalability: Easily scalable by adding more nodes at different levels.
- Organized Structure: Allows for better organization of large networks with various levels of hierarchy.
- Fault Isolation: Faults can be isolated to specific branches, preventing them from affecting the entire network.

Disadvantages

- Complexity: More complex to configure and maintain compared to other topologies.
- High Cost: Requires more cabling and is generally more expensive due to its complexity.
- Maintenance: Difficult to troubleshoot due to its size and layered structure.

(½ Mark each for any one correct advantage of Star and Tree topology)
(½ Mark each for any one correct disadvantage of Star and Tree topology)

OR

a) i. SMTP - Simple Mail Transfer Protocol
ii. TCP/IP - Transmission Control Protocol/Internet Protocol

b)

Feature	Web Browser	Web Server
Role	**Client:** Initiates requests for web content.	**Server:** Responds to requests and sends content.
Function	Displays web pages by interpreting code like HTML, CSS, and JavaScript.	Stores website files and delivers them to browsers.
Location	Runs on a user's device, such as a computer or smartphone.	Runs on a dedicated computer or cloud infrastructure.
Examples	Google Chrome, Mozilla Firefox, Safari.	Apache, Nginx, Microsoft IIS.

a) (½ Mark each for writing correct expansion)
b) ***(1 mark for any one correct difference)***

Section - C (3x3 = 9 marks)

29 3

```
def Display():
    f=open("city.txt",'r')
    line=f.readline()
    while line:
        b=line.split()
        if b[0][-1].upper() not in 'AEIOU':
                print(b[0], b[1])
        line=f.readline()
```

(½ Mark for opening the file in default/correct mode)
(1 Mark for correct reading & iteration)
(1 Mark for correct logic of extracting the city not ending with a vowel)
(½ Mark for displaying city with STD code)

OR

```
def countlines():
    f=open("STORY.TXT",'r')
    a=f.readlines()
    print(a)
    c=0
    for line in a:
        if line[-2].upper() == 'R':
            c+=1
    return c
d=countlines()
```

```
print(d)
```

(½ Mark for opening the file in default/correct mode)
(½ Mark for correct reading & iteration)
(½ Mark for correct logic to process each word)
(1 Mark for correct condition and counting)
(½ Mark for returning the count)

30 CITY=[ ] 3

i)
```
def PUSH_CITY(CITY, new_city):
    CITY.append(new_city)
```

ii)
```
def POP_CITY(CITY):
    if len(CITY)==0:
        print("Underflow")
    else:
        return(CITY.pop())
```

iii)
```
def DISPLAY(CITY):
    if len(CITY)==0:
        print("None")
    else:
        for i in range(-1,-len(CITY)-1,-1):
            print(CITY[i])
```

(i) (1 Mark for correct definition of PUSH_CITY(CITY, new_city)
(ii) (½ Mark for correctly checking and displaying "Underflow")
(½ Mark for correctly popping and returning popped tuple/data)
(iii) (½ Mark for correctly checking whether the Stack is Empty or not)
(½ Mark for correctly printing the stack top to bottom)

31 4.5#6.0 3
7.0 @
4.5#@

OR

15@
7@
9@

(1 Mark for writing each correct line of output)

Section - D (4x4 = 16 marks)

32 Consider the table **Graduate** 4

SNO	NAME	STIPEND	SUBJECT	AVG	DIV
1	KARAN	400	PHYSICS	68	I

2	DIWAKAR	450	COMP Sc	68	I
3	DIVYA	300	CHEMISTRY	62	I
4	REKHA	350	PHYSICS	63	I
5	ARJUN	500	MATHS	70	I
6	SABINA	400	CHEMISTRY	55	II
7	JOHN	250	PHYSICS	64	I
8	ROBERT	450	MATHS	68	I
9	RUBINA	500	COMP Sc	62	I
10	VIKAS	400	MATHS	57	II

1. SELECT NAME
   FROM Graduate
   WHERE DIV='I'
   ORDER BY NAME;

   (½ Mark for correctly writing SELECT NAME FROM GRADUATE)
   (½ Mark for correctly writing WHERE DIV='I' ORDER BY NAME)

2. SELECT NAME, STIPEND, SUBJECT, STIPEND *12
   FROM Graduate;

   (½ Mark for correctly writing NAME, STIPEND, SUBJECT, STIPEND *12)
   (½ Mark for correctly writing FROM Graduate)

3. SELECT COUNT(*)
   FROM Graduate
   WHERE SUBJECT IN ('PHYSICS', 'COMPUTER SC');

   (½ Mark for correctly writing SELECT COUNT(*) FROM Graduate)
   (½ Mark for correctly writing WHERE SUBJECT IN ('PHYSICS', 'COMPUTER SC'))

4. INSERT INTO Graduate Values(11,"KAJOL", 300, "computer sc", 75, 1)

   (½ Mark for correctly writing INSERT INTO Graduate)
   (½ Mark for correctly writing Values(11,"KAJOL", 300, "computer sc", 75, 1))

OR

1. MIN(AVERAGE)

   63
   ---

2. SUM(STIPEND)

```
------------------------
        800
------------------------
```

3. AVG(STIPEND)

```
------------------------
        68.5
------------------------
```

4. COUNT(distinct SUBJECT)

```
------------------------
        4
------------------------
```

(1 Mark each for writing correct output)

33 i) 4

```
import csv
def disp():
        f=open("Employee.csv","r")
        records=list(csv.reader(f))
        for data in records:
            if int(data[2])>50000:
                print(data)
         f.close()
```

ii)

```
import csv
def countrec():
        f=open("Employee.csv","r")
        records=list(csv.reader(f))
        print(len(records))
        f.close()
```

(i)
(½ Mark for opening the csv file in correct mode
(½ Mark for reading the records from csv file)
(½ Mark for iteration of records)
(½ Mark for checking and displaying the matched records correctly)

(ii)
(½ Mark for opening the csv file in correct mode)
(½ Mark for reading the records from csv file)
(1 Mark for finding/counting & displaying the number of records)

34 Consider the following tables FACULTY and COURSES. Write SQL commands for the statements (1) to (4) 4

FACULTY

F_ID	Fname	Lname	Hire_date	Salary
102	Amit	Mishra	12-10-1998	12000
103	Nitin	Vyas	24-12-1994	8000

104	Rakshit	Soni	18-5-2001	14000
105	Rashmi	Malhotra	11-9-2004	11000
106	Sulekha	Srivastava	5-6-2006	10000

COURSES

C_ID	F_ID	Cname	Fees
C21	102	Grid Computing	40000
C22	106	System Design	16000
C23	104	Computer Security	8000
C24	106	Human Biology	15000
C25	102	Computer Network	20000
C26	105	Visual Basic	6000

1. SELECT * FROM FACULTY WHERE SALARY>12000;

   (½ marks for SELECT * FROM FACULTY)

   (½ marks for WHERE SALARY>12000)
2. SELECT * FROM COURSE WHERE FEES BETWEEN 15000 AND 50000;

   (½ marks for SELECT * FROM COURSE)

   (½ marks for WHERE FEES BETWEEN 15000 AND 50000)
3. UPDATE COURSES SET FEES=FEES+500 WHERE CNAME=’SYSTEM DESIGN’;

   (½ marks for UPDATE COURSES SET FEES=FEES+500)

   (½ marks for WHERE CNAME=’SYSTEM DESIGN’)
4. SELECT * FROM COURSES NATURAL JOIN FACULTY WHERE FNAME=’SULEKHA’ ORDER BY CNAME DESC;

   (½ marks for SELECT * FROM COURSES NATURAL JOIN FACULTY)

   (½ marks for WHERE FNAME=’SULEKHA’ ORDER BY CNAME DESC)

35 4

```
import mysql.connector as ms
def Add_display() :
   try:
       mydb=ms.connect(host='localhost', user='root', password ='root', database =
'SCHOOL', charset='utf8' )
      mycursor=mydb.cursor()
      ch='y'
      while ch.upper()=='Y':
         rno=int(input("Enter Roll Number : "))
         name=input("Enter Name")
         stream=input("Stream")
         fee=float(input("Enter Fee"))
```

```
            Query="insert into student values(%s,'%s','%s',%s)"%(rno, name, stream,
fee)
             # Query = "insert into student values({},'{}','{}',{})".format(rno, name,
stream, fee)
          mycursor.execute(Query)
          ch=input("Do you want to add more records?")
       mydb.commit()
       mycursor.execute("select * from student where fee<7000")
       rec=mycursor.fetchall()
       for row in rec:
          print(row)
       mydb.close()
   except ms.Error as E:
       print(E)
Add_display()
```

(1 Mark for creating correct connectivity)
(½ Mark for creating the cursor)
(½ Mark for correct formation of Query)
(½ Mark for correct execution of the query)
(½ Mark for correctly using commit())
(½ Mark for correct query to extract all the records)
(½ Mark for correct fetchall() and display all records)

Note: (½ Mark for importing correct module, if the marks allocated are less than 4)

Section - E (2x5 = 10 marks)

36 (I) 5

```
import pickle
def CreateFile():
   f=open("Book.dat", "wb")
   Bno=int(input("Enter Book No."))
   Bname=input("Enter Book Name")
   Aname= input("Enter Author Name")
   price=float(input("Enter Price"))
   rec=[Bno,Bname,Aname,price]
   pickle.dump(rec,f)
   f.close()
```

(II)

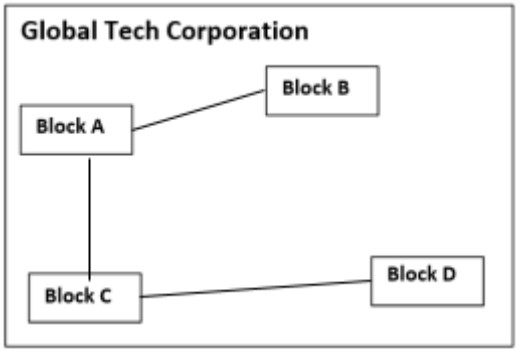
```
def CountRec(Author):
   cnt=0
   f=open("Book.dat","rb")
   try:
      while True:
         rec=pickle.load(f)
         if rec[2].upper()==Author.upper() :
            cnt+=1
   except EOFError:
      print("End of file reached")
```

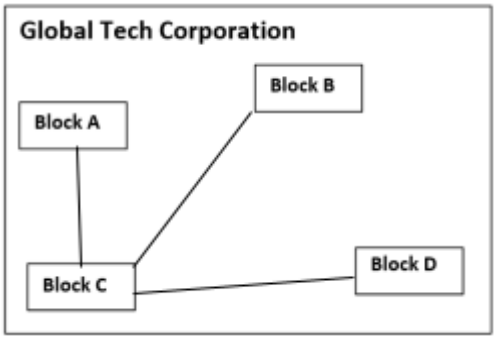
```
        return cnt
    f.close()
```

(i)
(½ Mark for opening the file in correct mode)
(½ Mark for accepting from user)
(½ Mark for converting the data into list)
(½ Mark for writing on file)

(ii)
(½ Mark for opening the file in correct mode)
(1 Mark for reading each record in loop (loop + load))
(½ Mark for correctly checking the condition)
(½ Mark for correctly counting the matching author)
(½ mark for correctly returning / displaying the count)

37 (a) **Option-1 (Shortest Cable Length)** **Option-2 (Placement of Server)** 5





(1 Mark for drawing the correct cable layout)

(b) **Repeater –**

In my **layout option-1**, Repeater is required between Block C and D as distance is !00 Mtr.

In my **layout option-2**, Repeater is required between Block C and D, Block C and D as distance is !00 Mtr or more.

Hub/Switch – In every block to connect all the computers.

(½ Mark each for writing the correct placement of each device)

(c) Block C, as it has maximum number of Computers.

(1 Mark for suggesting a block with valid justification)

(d) Satellite Link (Microwave)

(1 Mark for writing the correct answer)

(e) VoIP (Voice over Internet Protocol) protocols are used to convert voice calls into digital data packets and transmit them over an IP network, allowing voice and video communication over the internet instead of traditional phone lines.

OR

WAN

(1 Mark for writing the correct answer)