Kendriya Vidyalaya Sangathan, Jaipur Region
Practice paper-3

Class: XII **Subject**: Computer Science (083)
Maximum Marks: 70 **Period**: 3 Hours

Instructions:

- This question paper contains 37 questions.
- All questions are compulsory. However, internal choices have been provided in some questions. Attempt only one of the choices in such questions
- The paper is divided into 5 Sections- A, B, C, D and E.
- Section A consists of 21 questions (1 to 21). Each question carries 1 Mark.
- Section B consists of 7 questions (22 to 28). Each question carries 2 Marks.
- Section C consists of 3 questions (29 to 31). Each question carries 3 Marks.
- Section D consists of 4 questions (32 to 35). Each question carries 4 Marks.
- Section E consists of 2 questions (36 to 37). Each question carries 5 Marks.
- All programming questions are to be answered using Python Language only.
- In the case of MCQ, the text of the correct answer should also be written.

Q.	Section-A (21 x 1 = 21 Marks)	Mark
1	False	1
2	b) [‘Comma(’, ‘) is a punctuator’]	1
3	c) True or not True and False	1
4	a) 'ICSP'	1
5	a) 'nde'	1
6	a) statement 4	1
7	b) item.update((‘Patties’,30))	1
8	c) removes and return element at index x	1
9	c) 1,4	1
10	file.seek(0,2)	1
11	True	1
12	b) 5 5	1
13	Alter command	1
14	d) Details of all employees with two a’s together only in the name.	1
15	d) (a) & (c)	1
16	a) 950	1
17	b) POP3	1
18	d) Modem	1
19	Circuit Switching	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	(B) Both A and R are true and R is not the correct explanation for A	1
21	(D) A is False but R is True	1

<table>
<tr><th>Q</th><th>Section-B (7 x 2=14 Marks)</th><th>Mark</th></tr>
<tr><td>22</td><td>s[0] = ‘g’ will result in error because string is immutable data type and does not support item assignment.</td><td>2</td></tr>
<tr><td>23</td><td><table><tr><td>Type</td><td>Operator</td></tr><tr><td rowspan="2">Arithmetic</td><td>**</td></tr><tr><td>/ // % *</td></tr></table></td><td>2</td></tr>
</table>

	+ -
Logical	not
	and
	or

24	(i) (A) A.count(10) OR (B) A.sort() (ii) (A) A.extend(B) OR (B) B.sort(reverse=True) *(1 mark each)*	2
25	Minimum value of r – 1 Maximum value of r – 3 a) 30#40#50#60#70# d) 40#50#60# *(1/2 mark each for min and max value & ½ mark each for correct output)*	2

26	def lshift(t,n):
	    n=n%len(t)
	    t=t[n:]+t[:n] # n in place of n+1
	    return t # return indentation is removed
	
	newt=lshift((1,2,3,4,5,6),13)
	print(newt)
	(1 mark for each correction)
		2

27	(i) (A) Unique OR (B) Default (ii) (A) ALTER TABLE STUDENT DROP PRIMARY KEY; OR (B) ALTER TABLE STUDENT ADD PRIMARY KEY(RNO); *(1 mark each)*	2
28	Advantage: Hierarchical connection between the nodes. Disadvantage: Less reliable than star and mesh. *(1 mark each)* OR TELNET : Teletype Network Telnet is a network protocol used to virtually access a computer and provide a two-way, collaborative and text-based communication channel between two machines. It follows a user command TCP/IP networking protocol that creates remote sessions. *(1 mark for expansion and 1 mark for use)*	2

Q	Section-C (3 x 3 = 9 Marks)	Mark

29	(A)
	def govWeb():
	    f=open("URLs.txt", 'r')
	    data=f.read()
	    low=data.split()
	    for w in low:
	        if 'gov.in' in w:
		3

```
            print(w)
    f.close()
```

(1/2 mark for correct function definition)

(1/2 mark for opening file)

(1/2 mark for reading data)
(1/2 mark for correct iteration)
(1/2 mark for correct if)
(1/2 mark for correct print)

OR

(B)

```
def atleast5():
    f=open("Story.txt", 'r')
    data=f.read()
    low=data.split()
    for w in low:
        if len(w)>=5:
            print(w)
    f.close()
```

(1/2 mark for correct function definition)

(1/2 mark for opening file)

(1/2 mark for reading data)
(1/2 mark for correct iteration)
(1/2 mark for correct if)
(1/2 mark for correct print)

30 (A) 3

```
def push_star(StarStudent, AllStudents):
    for i in AllStudents:
        if i['marks']>90:
            StarStudent.append(i)

def pop_star(StarStudent):
    if StarStudent:
        return StarStudent.pop()
    else:
        print("Underflow")

def peek_star(StarStudent):
    if StarStudent:
        return StarStudent[-1]
    else:
        print("None")
```

(1 mark for each correct function definition)

OR

(B)

```
pos_int=[ ]
def push_positive(N):
    for i in N:
        if i>0:
            pos_int.append(i)

def pop_positive():
    if pos_int:
        return pos_int.pop()
    else:
        print("Empty")

def disp_positive():
    for i in range(-len(pos_int),0,-1):
        print(pos_int[i], end=“ ”)
    else:
        print("None")
```

(1 mark for each correct function definition)

31 Shoes10#
Gloves20#
Jackets15#

Mark: 3

(1 mark for each correct output)

OR

```
1 #
4 #  3 #  2 #  1 #
2 #  1 #
2 #  1 #
4 #  3 #  2 #  1 #
1 #
```

(1/2 mark for each correct output)

Q	Section-D (4 x 4 = 16 Marks)	Mark

32 (A)

Mark: 4

(i) SELECT SUM(PRICE) FROM EVENTS GROUP BY E_NAME HAVING SUM(PRICE) < 100000;
(ii) SELECT * FROM EVENTS ORDER BY CAPACITY DESC;
(iii) SELECT DISTINCT(E_NAME) FROM EVENTS;
(iv) SELECT SUM(PRICE) FROM EVENTS WHERE CAPACITY IS NOT NULL

(1 mark for each correct query, ½ for partially correct)

OR

(B)

(i)

E_name	sum(price)
Birthday	6500
Anniversary	15000
Reception	25000

(ii)

Manager
Prateek
Manoj
Shivansh

(iii)

e_id	Price
1001	3000
1004	3500

(iv)

max(price)
25000

(1 mark for each correct output, ½ for partially correct)

33 CSV file based question

Mark: 4

(i)

```
import csv
f=open("Employment.csv")
```

```
ro=csv.reader(f)
for rec in ro:
    if rec[1]>5000000:
        print(rec)
```

(1/2 mark for correct file opening)
(1/2 mark for reader object)
(1/2 mark for reading data)
(1/2 mark for correct if and print)

(ii)

```
import csv
f=open("Employment.csv")
ro=csv.reader(f)
l=list(ro)
print("no. of records = ", len(l))
```

(1/2 mark for correct file opening)
(1/2 mark for reader object)
(1/2 mark for reading data)
(1/2 mark for correct print)

34 — 4

```
(i) SELECT * FROM STUDENT, CLUB WHERE MARKS<80 AND
    STUDENT.CLUBID = CLUB.CLUBID;
(ii) SELECT * FROM CLUB WHERE FEES BETWEEN 400 AND 700;
(iii) UPDATE CLUB SET FEES=FEES+200 WHERE CNAME LIKE ‘%O’;
(iv) SELECT NAME, MARKS FROM STUDENT, CLUB WHERE CNAME=’CYBER’
      AND STUDENT.CLUBID = CLUB.CLUBID;
OR
SELECT * FROM STUDENT NATURAL JOIN CLUB;
```

(1 mark for correct query)

35 — 4

```
def addRec():
    import mysql.connector as m
    con=m.connect(host='localhost', user='root',  passwd='Chetan',
          database='SHOP')
    cur=con.cursor()
    ino=int(input("Enter item no."))
    iname=input("Enter item name")
    p=float(input("Enter price"))
    q=int(input("Enter quantity"))
    cur.execute("insert into inventory values(%s,%s,%s,%s)", [ino,iname,p,q])
    con.commit()
    cur.execute("select * from inventory where price>150")
    for rec in cur.fetchall():
        print(rec)
```

(1/2 mark for correct connection)
(1/2 mark for cursor)
(1/2 mark for correct intput)
(1/2 mark for correct insert query execution)
(1/2 mark for commit)
(1/2 mark for correct select query execution)
(1/2 mark for correct fetch and iteration)
(1/2 mark for correct print)

Q	Section-E (2 x 5 = 10 Marks)	Mark

36 — 5

```
def addBin():
    import pickle
    f=open('Students.dat', 'ab')
    cid=int(input("Enter candidate id :"))
```

	cname=input("Enter candidate name")
	    c=input("Enter class")
	    s=input("Enter status(active/passed out)")
	    pickle.dumpt([cid,cname,c,s],f)
	    f.close()
	(1/2 mark for correct file open)
	(1/2 mark for correct input)
	(1/2 mark for correct dump)
	def promoteBin():
	    import pickle
	    f=open('Students.dat', 'rb+')
	    try:
	        while True:
	            p=f.tell()
	            d=pickle.load(f)
	            if d[2]=='xii':
	                d[3]='passed out'
	                f.seek(p)
	                pickle.dump(d,f)
	    except:
	        f.close()
	(1/2 mark for correct file open)
	(1/2 mark for correct iteration and read)
	(1/2 mark for correct if and modification)
	(1/2 mark for correct seek and dump)
	def activeBin():
	    import pickle
	    f=open('Students.dat', 'rb')
	    try:
	        while True:
	            d=pickle.load(f)
	            if d[3]!='passed out':
	                print(d)
	    except:
	        f.close()
	(1/2 mark for correct file open)
	(1/2 mark for correct iteration and read)
	(1/2 mark for correct if and print)	

37	(a) Star with ADMIN as center (b) ADMIN block due to maximum number of computers in any block because of which most of the traffic will be local. (c) i) Switch/Hub: In every block which has more than 1 device to interconnect them ii) Repeater: Between ADMIN and SALES to regenerate the weak signal as distance is over 80m (d) Ethernet cable (e) a) WAN OR b) (i) Video conferencing *(1 mark for each correct answer)*	5