PRACTICE PAPER – 2 (SOLUTION)

CLASS : XII

SUBJECT: COMPUTER SCIENCE

SECTION-A

1. True

2. (B) 1st_string

3. (C) indentation

4. (C) Logical error

5. (A) Run time error

6. (B) tuple

7. (D) /

8. (B) COUNT()

9. (A) def

10. (D) Primary key

11. (A) \n

12. (B) cursor.fetchall()

13. (B) regenerate

14. (A) STACK

15. (D) Tree

16. (C) FIELD

17. (D) A is false but R is True

18. (A) Both A and R are true and R is the correct explanation for A

SECTION-B

19. Value=30

```
for val in range(0, Value) : # Error 1
        of val%4 = =0: # Error 2
                print (val*4)
        elif val%5 = = 0: # Error 3
                print (val+3)
        else: # Error 4
                print(val+10)
```

20. Bps, Kbps, Mbps, Gbps, Tbps

OR

(i) Microwave

(ii) Radio wave

21. (a) [10, 8.91, 'Apple']

(b) This code will produce an error as in line 2 there is no value of 'a' available to be used in RHS of the assignment statement .

22. **Candidate keys:** Id, Product

Primary key: Id

23. **(a)** Cookies

(b) A server manages network resources in a network.

24. (ii) 30#40#50#

Maximum value for FROM,TO are 3,4 respectively.

OR

25*50!100!25

25. The wildcard operators are used with the LIKE operator to search a value similar to a specific pattern in a column. There are two wildcard operators.

% - represents 0,1 or many characters

- - represents a single number or character

OR

```
import MySQLdb
db = MySQLdb.connect('localhost','Hos Admin','HosAdmin@123','XYZHospital')
cursor=db.cursor()
cursor.execute("DROP TABLE IF EXISTS DOCTORS")
sql="""Create Table Doctor ('Doctor_Id' INT NOTNULL, 'Doctor_Name' Char(50) NOTNULL,
'Hospital_Id' INT NOTNULL, 'Joining_Date' Date NOTNULL, 'Speciality' char(50), 'Salary'
        Float, 'Experience' Float
PrimaryKey ('Doctor_Id'))"""
cursor.execute(sql)
cursor.closet)
db. close ()
```

SECTION-C

26. **(a)** SELECT * FROM Teacher WHERE Department = "History";

(b) **(i)**

TID	TNAME
103	DEEPTI
106	MANIPRABHA

(ii)

Distinct TID
101
103
102
104
105

(iii)

TID	COUNT(*)	MIN (FEES)
101	2	12000

(iv)

COUNT (*)	SUM (FEES)
4	65000

27.

```
def countP ( ):
    fname = "DIARY.txt"
    with open (fname, 'r') as f:
        l=f.readlines()
        for line in l
            if l.strip=='p':
                print (line)
        f.close()
```

OR

```
def lowerupper() :
    fname = "Book.txt"
    lcount=ucount=0
    with open(fname,  'r') as f:
        for line in f:
            for word in line:
                for char in word;
                    if char.islower() :
                        1count=1count + 1
                    if   char.isupper():
                        ucount = ucount + 1
    print  ("No. of lower case letters=",1count)
    print   (MNo.   of upper case letters=",ucount)
```

28. **(a)** (i)

M_Id	SUM(M_Qty)
MB004	450
MB003	400
MB001	300
MB006	200

(ii)

MAX(M_Mf_Date)	**MIN(M_Mf_Date)**
2017-11-20	2010-08-21

(iii)

M_Company	**M_Price**
Sony	7500
OPPo	8500

(iv) 5450

(b) UPDATE

29.

```
def overlapping   (list1,   list2) :
        l1=len(list1)
        l2=len(list2)
        for    i in range  (11):
              for j in range  (12):
              if listl [i] ==list2 [j] :
        return True
        return False
```

30.

```
books=[ ]
def Addnew(books,name) :
    books.append (name)
    print("bookl:",name,"inserted")
def Remove(books):
    if   books== [ ] :
        print("stack is empty")
    else:
        print("book:",books.pop( ),"deleted ")
```

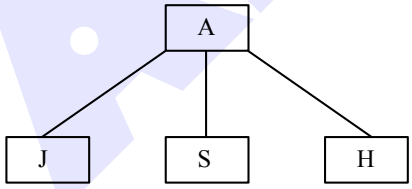
OR

(i)
```
def PUSH  (Arr,  value) i
    s = [ ]
    for x in range  (0, len(Arr)):
        if Arr[x] %5==0:
            s.append (Arr[x])
        if   len(s)==0:
            print ("EMPTY Stack")
        else:
            print(s)
```

(ii)
```
def popstack(st) :  #if stack is empty
    if len(st) == NULL :
        print ("Underflow")
    else:
        l=len(st)
        val=st(1-1)
        print(val)
        st.pop(1-1)
    return val
```

SECTION-D

31. (i)



(ii) Server can be placed in the A building as it has the maximum number of computers.

(iii) Repeater can be placed between A and S and A and J buildings as the distance is more than 110 m. Hub/switch should be placed in each building.

(iv) Radiowaves can be used in hilly regions as they can travel through obstacles.

(v) LAN (Local Area Network) as this network can be carried out within 1 km network.

32. (a) fUN#pYTHONn#

(b)
```
import MySQLdb
    db=MySQLdb.connect('localhost', 'Admin',  'Admin@pwd', 'cosmetics')
    cursor=db.cursor( )
    sql=" " "select * from Product where Price BETWEEN %F to %F"""
    value = (50, 100)
    try:
    cursor.execute(sql, value)
    results = cursor.fetchall( )
    print ("ID ProductName Manufacturer Price")
    for row in results:
        PID = row[0]
        PName = row[1]
        PManu = row[2]
        PPrice = row[3]
        print (" %s %s %s %s", % \ (PID, PName, PManu, PPrice))
        print (cursor.rowcount, "Records Found")
    except:
        print ("Error unable to Fetch data")
        cursor.close ( )
    db.close( )
```

OR

(a) quit talking and begin doing

(b)
```
import MySQLdb
db = MySQLdb.connect('localhost', 'Admin', 'Admin@pwd', 'cosmetics')
cursor=db.cursor( )
sql="""select ClientName, City From Client"""
try:
    cursor.execute(sql)
    results = cursor.fetchall( )
    print  ("ClientName", "City")
```

```
            for row in results :
                print ("%s  %s",  % \ (row[0], row[l] ))
                print  (cursor.rowcount, "Records Found")
        except:
                    print ("Error unable to Fetch data")
                cursor.close( )
        db. close ( )
```

33. When open ( ) function is used. Python stores the reference of mentioned file in the file_object

PROGRAM

```
fpl = open ("phonebook.dat", 'w')
fpl.write  ("Name")
fpl.write  (" ")
fpl.write  ("Phone")
fpl.write  ('\n')
while True:
    name = input ("Enter name:")
    phno = input ("Enter phone no:")
    fpl.write (name)
    fpl.write (" ")
    fpl.write (phno)
    fpl.write ("\n")
    ch = input ("Want to enter more=y/n")
    if   ch == 'N' or ch ==  'n':
        break
fpl.close ( )
```

OR

closed attribute stores True or False depending on if file is closed or not.

PROGRAM

```
fpl = open ("phonegook. dat", 'w+')
list = " "
while list:
    pos . ftell( )
    list =  fpl.readline ( )
    name, phone = list.split ( )
    if name == "Arvind":
            phone = input ("Enter new number:")
        fp.seek (pos,0)
        fp.write (name)
        fp.write (" ")
        fp.write (phone)
        fp.close ( )
        break
    else :
        print ("Name\"Arvind\ "not found.")
```

SECTION-E

34. **(i)** values of ACode in table Activity

(ii) DESC

(iii) (a) SELECT ACode, ActivityName FROM ACTIVITY ORDER BY ACode DESC;

(b) SELECT SUM(PrizeMoney) FROM ACTIVITY GROUP BY Stadium;

(c) SELECT Name, ACode FROM COACH ORDER BY ACode;

OR

(iii) (a) 3

(b) 19-Mar-2004 12-Dec-2003

35. **(a)** csv

(b) "Employee.csv","r"

(c) for R in ER:

ALLEN®