

COMPUTER SCIENCE

SECTION-A

1. False

Explanation : The update () method updates a dictionary contents by another.

2. (B)

Explanation : The order by clause is used to arrange records of a table either ascending order or descending order.

3. (C)

Explanation : $711\ 14 + 12\% \ 5 + 9^{**}2 - 1$

$$= 18 + 2 + 81 - 1$$

$$= 20 + 80$$

$$= 100$$

4. (C)

Explanation : str (m) returns "1" , which is concatenated to n="1" to give 11

5. (B)

Explanation : Database schema is the logical representation of data which shows how the data is stored logically in the entire database.

6. (D)

Explanation : Bluetooth is a short distance wireless transmission.

7. (D)

Explanation : p+q is a string concatenation, r+s is a normal addition

8. (A)

Explanation : The nested tuple has (4,3, (1, 3,2)) as the element at index 2. Again(1, 3,2) is the element at index 2 in this tuple.

9. (C)

Explanation : The split () function splits a string on a certain character.

10. (B)

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

11. (A)

Explanation: Cookies are temporary internet files that store user information

12. (B)

Explanation: The random () function returns a random number between 0-1

13. (B)

Explanation: The code written in the finally block executes, irrespective of whether exception occurs or not.

14. (A)

Explanation : A table can have only a single primary key.

15. Amplify

Explanation : A repeater is a network device that amplifies signals, so that they can travel a larger distance.

16. (A)

Explanation : The 'r+' opens a file for both reading and writing.

17. (D)

Explanation : Indexing is used for accessing individual characters within a string. The first character has the index 0 and the last character has the index n-1 where n is the length of the string. The negative indexing ranges from -n to -1.

Strings in Python are immutable, i.e., a string cannot be changed after it is created.

18. (A)

Explanation : The concatenation operator + when used with two lists, joins them. The + operator when used with lists requires that both operands must be of list types. You cannot add any other value to a list.

Example:

```
>>>list1=[1,2,3]
```

```
>>>list2 = [4,5,6]
```

```
>>>list1 +list2
```

```
[1,2,3,4,5,6]
```

SECTION-B

19. (a) **POP3 - Post Office Protocol 3**

URL - Uniform Resource Locator 2

(b) **HTML(Hyper text MarkUp Language)**

- We use pre-defined tags
- Static web development language -only focuses on how data looks
- It use for only displaying data, cannot transport data
- Not case sensitive
- XML (Extensible Markup Language)
- We can define our own tags and use them
- Dynamic web development language - as it is used for transporting and storing data
- Case sensitive

OR

(a) Bandwidth is the maximum rate of data transfer over a given transmission medium.

(b) https (Hyper Text Transfer Protocol Secure) is the protocol that uses SSL (Secure Socket Layer) to encrypt data being transmitted over the Internet. Therefore, https helps in secure browsing while http does not.

20. **def** revNumber (num) :

rev = 0

rem = 0

while num > 0:

rem = num %10

rev = rev*10 + rem

num = num//10

return

print (revNumber(1234))

Explanation: The keyword to define a function is "def" not "define"

The k word is "while" not "While"

The return statement should be after the end of the while loop. So indentation should be done accordingly.

21. PLACES = {1: "Delhi", 2: "London", 3: "Pariaos", 4: "New York", 5: "Dubai"}

```
def countNow (PLACES):
    for place in PLACES. Values ():
        if len (place) > 5:
            print (place. upper())
countNow (PLACES)
```

OR

```
def lenWords (STRING):
    T = ()
    L = STRING, split ()
    for word in L:
        length = len (word
    T = T + (length,)
    return T
```

Explanation: 1st option code:

The for loop traverses through the values of the dictionary and checks them if their length is more than 5. In these cases the values are converted to uppercase and printed.

2nd option code:

The split() function parses the string into words and returns a list L carrying each word as a list item. The for loop picks each word from the list and finds the length of each. It then adds the length found as an element of the tuple.

Note: Any other correct logic may be marked

22. 4*L
33*4
21*S
10*6

Explanation:

S="LOST"

L=[10,21,33,4]

D={}

for I in range(len(S)):

if I%2==0:

D[L.pop()]=S[I]

else:

D[L.pop()]=I+3

for K,V in D.items():

print(K,V,sep="*")

In the code, the for loop traverses through the length of the string that is, I picks values from 0 to 3. If I is even the statement `D[L.pop()]` pops the last element from the list L and adds it as a dictionary key. The value of the key is taken as `S[I]` that is character at index I of the string S.

In the output for loop, each dictionary key and values is picked from `D.items()` and printed with a separator `"*"`

23. (a) `L1.insert(2,200)`
 (b) `message.endswith('.')`

Explanation: The `insert()` function inserts an element at the index specified.

The `ends with()` function checks whether a string ends with a specific string or not and returns True/ False.

OR

```
import statistics
print( statistics.mode(studentAge))
```

24. Part 1:

```
ALTER TABLE Employee ADD EmpId INTEGER PRIMARY KEY;
```

Part 2:

```
INSERT INTO Employee VALUES("Shweta","Production",26900,999);
```

OR

```
INSERT INTO Employee(EmpId,Ename, Department, Salary)
VALUES(999,"Shweta","Production", 26900);
```

Explanation: The Alter table command can be used to add, modify and remove a field. It can also be used to add a constraint.

Syntax: `ALTER TABLE <tablename> ADD <Field> <Type> PRIMARY KEY;`

The syntax to add a record to a table is: `Insert into <table> values (val1,val2,val3,...)`

OR

To delete the attribute, category:

```
ALTER TABLE Sports DROP category;
```

To add the attribute, TypeSport

```
ALTER TABLE Sports ADD TypeSport char(10)
```

```
NOT NULL;
```

25. Output

```
10.0$20
```

```
10.0$2.0###
```

```
def Changer (P, Q = 10):
```

```
    P = P/Q
```

```
    Q = P%Q
```

```
    return P
```

```
A = 200
```

```
B = 20
```

```
A = Changer (A, B)
```

```
print (A, B, sep = ' $')
```

```
B = Changer (B)
```

```
Print (A, B, sep = '$', end = '###')
```

Explanation: The Changer function receives a value P and a default argument in the form of Q=10. In the first call the values received in P, Q are 200,20. The statement $P=P/Q$ makes $P = 200/20 = 10.0$ and $Q=p\%Q = 10.0\%10 = 0$. The value of P is returned back to $A=10.0$.

The print statement prints values of A, B which are 10.0 and 20 with a separator \$. Similarly in the 2nd call the values received by P and Q are 20,10(The default value). $P=P/Q$ makes P as 2.0. Value of $P=2.0$ is returned into B. Hence the values printed are 10.0, 2.0 with separator '\$' and ending with "###"

SECTION-C

26. SELCcME&Cc

27. (i) COUNT (*) City

2 Mumbai

2 Delhi

2 Moscow

(ii) MAX (DOB) MIN (DOB)

08-2-1995 07-05-1993

(iii) NAME GENDER

Sonal F

Store M

```
28. def COUNTLINES () :
    file = open ('TESTFILE.TXT', 'r')
    lines = file.readlines()
    count = 0
    for w in lines :
        if (w[0].lower() not in 'aeoiu'):
            count = count + 1
    print("The number of lines not starting with any vowel:", count)
    file.close()
    COUNTLINES()
```

OR

```
def ETCount():
    file=open('TESTFILE.TXT','r')
    lines = file.readlines()
    countE=0
    countT=0
    for w in lines :
        for ch in w:
            if ch in 'Ee':
                countE=countE + 1
            if ch in 'Tt':
                countT=countT + 1
    print("The number of E or e : ",countE)
    print("The number of T or t : ",countT)
    file.close()
```

```
29. 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")
```

SECTION-D

31. (i) Select Designation, Count (*) From Admin Group By Designation Having Count (*) <2;
- (ii) SELECT max (EXPERIENCE) FROM SCHOOL;
- (iii) SELECT TEACHER FROM SCHOOL WHERE EXPERIENCE >12 ORDER BY TEACHER;
- (iv) Select Teacher, Designation from School s, Admin a where a.code=s.code;

32. import csv

```
def ADD () :  
    fout=open("record.csv","a",newline="\n")  
    wr=csv.writer(fout)  
    empid=int(input("Enter Employee id :: "))  
    name=input("Enter name :: ")  
    sal=int(input("Enter salary :: "))  
    lst=[empid,name,sal]  
    wr.writerow(lst)  
  
fout.close()  
  
def COUNTRO :  
    fin=open ("record. csv","r",newline="\n")  
    data=csv. reader (fin)  
    d=list(data)  
    print(len(d))  
  
fin.close ()  
ADD()  
COUNTR()
```

SECTION-E

33. (i) Finance
- (ii) Star topology.
- (iii) Satellite Link.
- (iv) Switch
- (v) Broadband

34. (i) When open () function is used, Python stores the reference of mentioned file in the file_ object., (1 mark for correct answer)

(ii) **PROGRAM**

```
fpl = open ("phonebook.txt", '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()
```

(1 mark for correct

answer)

OR

- (i) Closed attribute stores True or False depending on if file is closed or not.

(ii) **PROGRAM**

```
tpl = open ("phonebook.txt", '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.")
```

35. (i) Join: A join is a relational operation that brings data from multiple tables by linking them on a common field.

(ii) `import MySQLdb`

```
db=MySQLdb.connect('localhost','Admin','Admin@123','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

(i) **SHOW CHARACTER SET;**

(ii) `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[1]))
    print (cursor.rowcount,"Records Found") except:
    print ("Error unable to Fetch data")
cursor.close()
db.close()
```

ALLEN[®]