

**SET –II MARKING SCHEME**  
**KENDRIYA VIDYALAYA GUWAHATI REGION**  
**CLASS: XII SESSION: 2024-25**  
**COMPUTER SCIENCE (083)**

**MARKING SCHEME/ANSWER KEY**

1.	False							
2.	(a) 'ast#ba'							
3.	(c) 18							
4.	(b) [' ', 'ython ', 'rogramming']							
5.	'dont oe'							
6.	(a) print(sum(t))							
7.	(c) print(mdict['black', 'white'])							
8.	(d) None of these							
9.	(b) <class 'tuple'>							
10.	(a) 0							
11.	True							
12.	(c) 50\$							
13.	Alter table SRecord Add primary key (ADNO);							
14.	Order By							
15.	(d) Cardinality							
16.	(b) Select count(*) from ORDERS;							
17.	(b) Hypertext Transfer Protocol Secure							
18.	(c) Hz							
19.	Transmission Control Protocol/ Internet Protocol							
20.	(a) Both A and R are true and R is the correct explanation for A							
21.	(b) Both A and R are true and R is not the correct explanation for A							
22.	f = <u>1</u> num = <u>int</u> (input("Enter a number: ")) n = num while num > 1 : f = f * num <u>num - = 1</u> else : print ("The factorial of: ", <u>n</u> , "is", f)	½ X 4 = 2						
23.	1\$4\$7\$10\$							
24.	(i) (A) L. extend(L1)/L1.extend(L) OR (B) L. sort () (ii) (A) SL = sorted (L1, reverse = True) OR (B) L. insert (5, L1[-1])	1 mark each						
25.	(A), (C) (½ x 2 = 1 Mark) Minimum and maximum possible values of the variable b: 1,6 (½ x 2 = 1 Mark)							
26.	<table border="1" style="width: 100%;"> <thead> <tr> <th style="width: 50%;">Primary Key</th> <th style="width: 50%;">Alternate key</th> </tr> </thead> <tbody> <tr> <td>1) Primary key is used to uniquely identify the records in a table.</td> <td>1) This is a candidate key other than the Primary key</td> </tr> <tr> <td>2) There can be only one primary key in a table/relation.</td> <td>2) There may be multiple alternate keys in a table/ relation.</td> </tr> </tbody> </table>	Primary Key	Alternate key	1) Primary key is used to uniquely identify the records in a table.	1) This is a candidate key other than the Primary key	2) There can be only one primary key in a table/relation.	2) There may be multiple alternate keys in a table/ relation.	
Primary Key	Alternate key							
1) Primary key is used to uniquely identify the records in a table.	1) This is a candidate key other than the Primary key							
2) There can be only one primary key in a table/relation.	2) There may be multiple alternate keys in a table/ relation.							

	Alternate Key	Primary Key		Alternate Key
	Emp_SSN	Emp_Id	Emp_name	Emp_email
	11051	01	John	john@email.com
	19801	02	Merry	merry@email.com
	19801	03	Riddle	riddle@email.com
	41201	04	Cary	cary@email.com

27

(i) (A) Alter table Marks  
Add column Percentage decimal (3,1);  
(B) Default constraint

(ii) (A) Alter table Product  
Change comm commission decimal (4,2);  
**OR**  
Alter table Product  
Rename column comm to commission;  
(B) Unique constraint

28

(i) Simple Mail Transfer Protocol  
(ii) Voice over Internet Protocol

29

```
(A) f = open("Para.txt",'r')
a = f.read ()
U=L=D=0
for i in a:
    if i.isupper():
        U += 1
    elif i.islower():
        L += 1
    elif i.isdigit ():
        D += 1
    else:
        continue
print ("Upper case alphabets - ", U)
print ("Lower case alphabets - ", L)
print ("Digits - ", D)
(B) f = open("DECODE.TXT",'r')
a = f.read ()
s = a.split ()
d = 0
for i in s:
    if len (i) >= 5:
        d += 1
    else:
        continue
print ("The words having 5 or more characters are ", d)
```

30

```
(A)
(i) def push_city (d_city):
    for i in d_city:
        if len (i) > 4:
            CITY.append (d_city[i])
(ii) def pop_city ():
    if len(CITY)==0:
        print ("Stack empty")
    else:
        print (CITY.pop())
(iii) def peep(d_city):
    if len (CITY)==0:
```

	<pre> print("None") else:     print (CITY[-1])  OR  (B) (I) def push_book (BooksStack, new_book):     BooksStack.append (new_book) (II) def pop_book (BooksStack):     if not BooksStack:         print("Underflow")     else:         return (BookStack.pop()) (III) def peep (BooksStack):     if not BooksStack:         print ("None")     else:         print (BookStack[-1]) </pre> <p><b>(3x1 mark for correct function body; No marks for any function header as it was a part of the question)</b></p>	
31	<p>(A)</p> <ul style="list-style-type: none"> <li>(i) Select color, count(*) from Rent_cab Group by color;</li> <li>(ii) Select * from Rent_cab Order by charges;</li> <li>(iii) Select Vcode, VName from Rent_cab Where make like '%o%';</li> </ul> <p>(B)</p> <ul style="list-style-type: none"> <li>(i) Update Rent_cab Set charges = charges + charges*10/100;</li> <li>(ii) Delete from Rent_cab Where make = 'Carus';</li> <li>(iii) Select color sum(charges) from Rent_cab Group by color;</li> </ul>	1 mark each
32	<p>A) i) An exception is said to be caught when a code that is designed to handle a particular exception is executed. Exceptions, if any, are caught in the try block and handled in the except block. While writing or debugging a program, a user might doubt an exception to occur in a particular part of the code. Such suspicious lines of codes are put inside a try block. Every try block is followed by an except block. The appropriate code to handle each of the possible exceptions (in the code inside the try block) are written inside the except clause.</p> <p>ii) print ("Using try block") try:     numerator=50     denom=int(input("Enter the denominator"))     quotient=(numerator/denom)     print ("Division performed successfully") except ZeroDivisionError:     print ("Denominator can not be ZERO") except :     print("Some other error occurred")</p> <p>B) i) IOError exception is raised when the file specified in a program statement cannot be opened.</p> <p>ii) try:     f=open("text.txt", 'r')     print("The content of file is ")     print(f.read())</p>	

	<pre> except IOError:     print ("File not found") except :     print("Some other error occurred") </pre>	
33	<pre> import csv def Add_detail():     f = open("sports.csv", 'a')     w= csv.writer(f)     sport_id = int(input("enter Sport id"))     competition = input("Competition")     prize_won = input("Prize won")     L = [sport_id, competition, prize_won]     w.writerow(L)     f.close()  def Count_Medal():     f = open("sports.csv", 'r')     L= csv.reader(f)     for i in L:         if i[2] == 'Gold':             print("Competition :", i[1])     f.close() </pre>	
34	<p>(i) Select type, Avg(Number) from Games Group by type;</p> <p>(ii) Select prizemoney, Gamename, name from Games, Players Where Games.GCode = Players.PCode;</p> <p>(iii) Select Distinct Type from Games;</p> <p>(iv) Select GameName , PrizeMoney from Games Where PrizeMoney is not null; OR Select * from Games, Players;</p>	
35	<pre> def ADRecord():     import mysql.connector as mycon     c = mycon.connect (host= "localhost", user= "Administrator", passwd= "market", database     ="Maintain")     cur = c. cursor()     I_code = int(input("Enter Item code: ") )     I_name = input("Enter Item Name: ")     Quan=int(input("Enter qty: "))     Amount = int(input("Enter price: "))     query="INSERT INTO Stall VALUES (i_code, i_name, Quan, Amount)"     cur. execute (query)     c.commit()     cur. execute ("select * from Stall where Amount between 200 and 300")     a = cur. fetchall()     for rec in a:         print(rec)     c.close() </pre>	
36	<p><b>Note: For part (I), the student can mention any type of file with valid reason to support the choice. Answer with valid supporting reason should be considered Correct, and without a valid reason should be considered incorrect.</b></p> <p>(I) Text file: A text file allows for easy maintenance of data, as it can be opened and manipulated with any text editor also. (1 mark for correct answer)</p> <p>(II) def append(): with open("record.txt",'a') as f: S_id=input("Enter student ID: ") S_nm=input("Enter student name: ")</p>	

```
S_fn=input("Enter Father's name: ")
S_per=input("Enter Percentage: ")
rec=S_id+','+S_nm+','+S_fn+','+S_per+'\n'
f.write(rec)
```

(½ mark for opening in the file in right mode)

(½ mark for correctly inputting the data)

(½ mark for correctly writing the record in the file)

(½ mark for correctly closing the file, or ½ mark if the file was opened using with)

(II) def display():

```
with open("record.txt") as f:
```

```
    for rec in f:
```

```
        data=rec.split(',')
```

```
        if float(data[-1])>85:
```

```
            print(rec.strip()) #OR print(rec)
```

(½ mark for opening the file in right mode)

(½ mark for correctly reading the data)

(½ mark for correctly checking the condition)

(½ mark for correctly displaying the records)

**OR**

(I) CSV File: A CSV file allows for easy maintenance of data, as it can be opened and manipulated with any spreadsheet application also.

(II) def append():

```
with open("record.csv",'a',newline=") as f:
```

```
    w=csv.writer(f)
```

```
    S_id=input("Enter student ID: ")
```

```
    S_nm=input("Enter student name: ")
```

```
    S_fn=input("Enter Father's name: ")
```

```
    S_per=input("Enter Percentage: ")
```

```
    rec=[ S_id,S_nm,S_fn,S_per]
```

```
    w.writerow(rec)
```

(III) def display():

```
with open("record.csv") as f:
```

```
    r=csv.reader(f)
```

```
    for rec in r:
```

```
        if float(rec[-1])>85:
```

```
            print(rec)
```

**OR**

(I) Binary File: A binary file cannot be opened and manipulated with any general purpose application, and hence, it prevents any unintentional change in the data.

(II) def append():

```
with open("record.dat",'ab') as f:
```

```
    S_id=int(input("Enter student ID: "))
```

```
    S_nm=input("Enter student name: ")
```

```
    S_fn=input("Enter Father's name: ")
```

```
    S_per=float(input("Enter Percentage: "))
```

```
    rec=[S_id, S_nm, S_fn, S_per]
```

```
    pickle.dump(rec,f)
```

(III) def display():

```
with open("record.dat",'rb') as f:
```

```
    while True:
```

```
        try:
```

```
            rec=pickle.load(f)
```

```
            if rec[-1]>85:
```

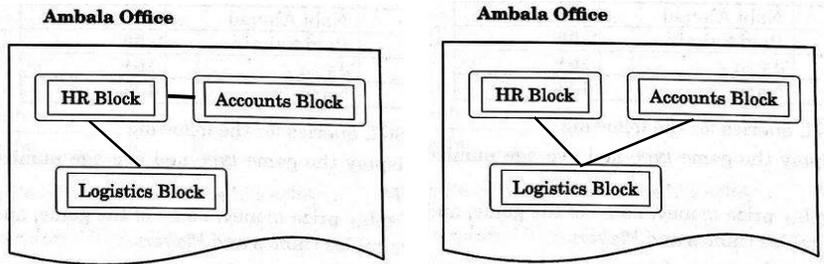
```
                print(rec)
```

```
        except EOFError:
```

```
            break
```

37

- (i) HR block as it has maximum number of computers.
- (ii) Optical Fiber
- (iii)



Connecting blocks with server

**OR** Connecting blocks with least distance

- (iv) VoIP
- (v) (A) WAN

**OR**

(B) Yes, The repeater is required as the distance is more than 70 meters between the blocks.