

THE DOON GLOBAL SCHOOL
PRE-BOARD EXAMINATION-2(2023-24)

CLASS - XII

SUBJECT- COMPUTER SCIENCE (083) Marking Scheme

Time: 3 Hours

Max. Marks: 70

SECTION A (1 mark each)

1. False
2. b) break
3. c) `tupl1 = tupl + tupl`
4. d) -1
5. a) `car["year"] = 2015`
6. a) PAN
7. b) `tell()`
8. a) HAVING
9. d) Statement 4
10. a) Error
11. c) Optical fiber
12. global b
13. c) global b
14. c) A candidate key that is not a primary key is a foreign key.
15. Circuit
16. c) `seek()`
17. a) Both A and R are true and R is the correct explanation for A.
18. d) A is false but R is true.

SECTION B (2 marks each)

19. i) POP3: Post Office Protocol 3 | URL: Uniform Resource Locator
ii) XML allows user-defined tags, HTML uses predefined tags.
OR
(i) Bandwidth is the maximum rate of data transfer over a given transmission medium or the amount of information that can be transmitted over a network.
(ii) HTTPS (Hyper Text Transfer Protocol Secure) is the protocol that uses SSL (Secure Socket Layer) to encrypt data being transmitted over the Internet.
20.

```
def revNumber(num): #correction 1
    rev = 0
    rem = 0
    while num > 0: #correction 2
```

```

        rem = num % 10 #correction 3
        rev = rev * 10 + rem
        num = num // 10
    return rev #correction 4
print(revNumber(1234))

```

21.

```

s="Come let us have some fun"
def lenWords(STRING):
    T = ()
    L = STRING.split()
    for word in L:
        length = len(word)
        T = T + (length, )
    return T
print(lenWords(s))

```

OR

```

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

```

22. Output:

```

4*L
33*4
21*S
10*4

```

23. Insert at position 3: L1.insert(2, 200)
Ends with period: message.endswith(".")

OR

from statistics import mode

24. ALTER TABLE Employee ADD EmpId INT PRIMARY KEY;
INSERT INTO Employee VALUES (999, 'Shweta', 'Production', 26900);

OR

ALTER TABLE Sports DROP COLUMN category;
ALTER TABLE Sports ADD TypeSport VARCHAR(50) NOT NULL;

25. Output: iNdiA%****

SECTION C (3 marks each)

26. NATURAL JOIN joins on common columns.
Corrected query: SELECT Dept_code, MAX(Salary) FROM EMPLOYEE GROUP BY Dept_code;
Virtual table: Result of a query treated as a table (view).

27. Check unique keys/values:

```

def check_unique(D):
    if len(D.values()) != len(set(D.values())):
        print("Values are not unique.")
    else:
        print("Values are unique.")

```

28. Count words in file:

```
def COUNT():
    with open("REPEATED.TXT") as f:
        text = f.read().lower()
        count = text.count("catholic") + text.count("mother")
    print("Count of Catholic, mother is", count)
```

AMCount function:

```
def AMCount():
    with open("STORY.TXT") as f:
        text = f.read().lower()
        print("A or a:", text.count("a"))
    print("M or m:", text.count("m"))
```

29. SQL queries:

- a) SELECT Name, Age FROM VACCINATION_DATA WHERE Dose2 IS NOT NULL AND Age > 40;
- b) SELECT City, COUNT(*) FROM VACCINATION_DATA GROUP BY City;
- c) SELECT DISTINCT City FROM VACCINATION_DATA;

30.

```
(i)
NList=[["New York", "U.S.A.", 11734], ["Naypyidaw", "Myanmar", 3219],
["Dubai", "UAE", 2194], ["London", "England", 6693], ["Gangtok", "India", 1580],
["Columbo", "Sri Lanka", 3405]]
travel = []
def Push_element(NList):
    for L in NList:
        if L[1] != "India" and L[2] < 3500:
            travel.append([L[0], L[1]])
    return travel
print(Push_element(NList))
(ii)
travel = ['Naypyidaw', 'Myanmar'], ['Dubai', 'UAE'], ['Columbo', 'Sri Lanka']
def Pop_element():
    while len(travel):
        print(travel.pop())
    else:
        print("Stack Empty")
Pop_element()
```

31. SQL Queries:

- a) SELECT PRODUCT.Pname, BRAND.Bname FROM PRODUCT JOIN BRAND ON PRODUCT.Brand_id = BRAND.Brand_id;
- b) DESC PRODUCT;
- c) SELECT Bname, AVG(Rating) FROM PRODUCT JOIN BRAND ON PRODUCT.Brand_id = BRAND.Brand_id WHERE Bname IN ('Medimix', 'Dove') GROUP BY Bname;
- d) SELECT Pname, Price, Rating FROM PRODUCT ORDER BY Rating DESC;

32. CSV Operations:

```
def Accept():
    sid = int(input("Enter Student ID"))
    sname = input("Enter Student Name")
    game = input("Enter name of game")
    res = input("Enter Result")
    headings = ["Student ID", "Student Name", "Game Name", "Result"]
    data = [sid, sname, game, res]
    f = open('Result.csv', 'a', newline = "")
    csvwriter = csv.writer(f)
    csvwriter.writerow(headings)
    csvwriter.writerow(data)
    f.close()
def wonCount():
    f = open('Result.csv', 'r')
    csvreader = csv.reader(f, delimiter = ',')
    head = list(csvreader)
    print(head[0])
    for x in head:
        if x[3] == "WON":
            print(x)
    f.close()
```

SECTION E (5 marks each)

33. Network Case Study:

- i) App Dev block centrally located
- ii) Firewall
- iii) Optical Fiber; layout from App Dev to others
- iv) Switches in each block, Repeater if >100m
- v) VoIP protocol like SIP or H.323

34. (i)

r+ mode	w+ mode
Primary function is reading.	Primary function is writing.
If the file does not exist, it results in an error.	If the file does not exist, it creates a new file.

ii)def copyData():

```
fobj = open("SPORT.DAT", "rb")
fobj1 = open("BASKET.DAT", "wb")
cnt = 0
try:
    while True:
        data = pickle.load(fobj)
        print(data)
        if data[0] == "Basket Ball":
            pickle.dump(data, fobj1)
            cnt += 1
except:
    fobj.close()
    fobj1.close()
return cnt
```

OR

(i)

Text files	Binary Files
Extension is .txt	Extension is .dat
Data is stored in ASCII format that is human readable.	Data is stored in binary form (0s and 1s), that is not human readable.

(ii)

```
def Searchtype(mtype):
    fobj = open("CINEMA.DAT", "rb")
    try:
        while True:
            data = pickle.load(fobj)
            if data[2] == mtype:
                print("Movie number:", data[0])
                print("Movie Name:", data[1])
                print("Movie Type:", data[2])

    except EOFError:
        fobj.close()
```

35. i) Domain is a set of values from which an attribute can take value in each row.

For example, roll no field can have only integer values and so its domain is a set of integer values.

ii) import mysql.connector as mysql

```
def sql_data():
    con1 = mysql.connect(host="localhost", user="root", password="tiger", database="school")
    mycursor = con1.cursor() # Statement 1
    rno = int(input("Enter Roll Number :: "))
    name = input("Enter name :: ")
    clas = int(input("Enter class :: "))
    marks = int(input("Enter Marks :: "))
    query = "INSERT INTO student VALUES({}, '{}', {}, {})".format(rno, name, clas, marks)
    mycursor.execute(query) # Statement 2
    con1.commit() # Statement 3
    print("Data Added successfully")
    con1.close()
```

OR

i) One difference between alternate key and candidate key:

Alternate Key	Candidate Key
A candidate key is a set of attributes (columns) in a table that uniquely identifies each record or row.	An alternate key is a set of attributes (columns) in a table that uniquely identifies each record but is not designated as the primary key.

ii) import mysql.connector as mysql

```
con1 = mysql.connect(host = "localhost", user = "root", password = "tiger", database = "SCHOOL")
mycursor = con1.cursor()
query = "SELECT * FROM student WHERE fee > {}".format(5000)
mycursor.execute(query)
data = mycursor.fetchall()
for rec in data:
    print(rec)
con1.close()
```