

ISC Paper 2017 Computer Science

Maximum Marks: 70
Time allowed: 3 hours

- Candidates are allowed additional 15 minutes for only reading the paper.
 - They must NOT start writing during this time.
 - Answer all questions in Part-I (compulsory) and six questions from Part-11, choosing two questions from Section-A, two from Section-B and two from Section-C.
 - All working, including rough work, should be done on the same sheet as the rest of the answer.
 - The intended marks for questions or parts of questions are given in brackets [].
-

Part – I (20 Marks) Answer all questions.

While answering questions in this Part, indicate briefly your working and reasoning, wherever required.

Question 1.

(a) State the law represented by the following proposition and prove it with the help of a truth table: $P \vee P = P$ [1]

(b) State the Principle of Duality. [1]

(c) Find the complement of the following Boolean expression using De Morgan's law: [1]

$$F(a, b, c) = (b' + c) + a$$

(d) Draw the logic diagram and truth table for a 2 input XNOR gate. [1]

(e) If $(\sim P \Rightarrow Q)$ then write its: [1]

(i) Inverse

(ii) Converse

Answer:

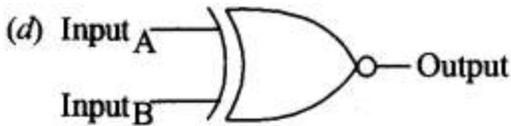
(a) The law represented by the given proposition is Idempotent law.

Truth Table:

| P | P | PVP | PVP \leftrightarrow P |
|---|---|-----|-------------------------|
| T | T | T | T |
| F | F | F | F |

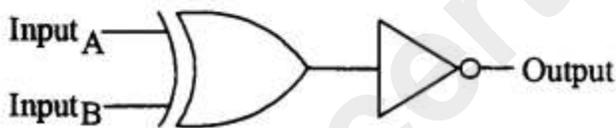
(b) The duality principle states that every algebraic expression deducible from the postulates of Boolean algebra remains valid if the operators and identity elements are interchanged.

$$\begin{aligned}
 (c) & ((b' + c) + a)' \\
 &= (b'+c)'. a' \\
 &= ((b')' . c') . a' \\
 &= bc'a'
 \end{aligned}$$



| A | B | Output |
|---|---|--------|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Equivalent gate circuit :



(e) (i) Inverse of the given statement

$$P \Rightarrow \sim Q$$

(ii) Converse of the given statement $Q \Rightarrow \sim P$

Question 2.

(a) What is an interface? How is it different from a class? [2]

(b) Convert the following infix expression to postfix form: [2]

$$P * Q/R + (S + T)$$

(c) A matrix $P[15][10]$ is stored with each element requiring 8 bytes of storage. If the base address at $P[0][0]$ is 1400, determine the address at $P[10][7]$ when the matrix is stored in Row Major Wise. [2]

(d) (i) What is the worst-case complexity of the following code segment: [2]

```

for(int x = 1; x<=a; x++)
{
statements;
}
for(int y = 1; y <= b; y++)

```

```

{
for (int z = 1; z <= c; z++)
{
statements;
}
}

```

(ii) How would the complexity change if all the three loops went to N instead of a, b and c?

(e) Differentiate between a constructor and a method of a class. [2]

Answer:

(a) An interface in Java is a blueprint of a class. It has static constants and abstract methods. The interface in java is a mechanism to achieve abstraction. There can be only abstract methods in the Java interface, not the method body. It is used to achieve abstraction and multiple inheritances in Java. It cannot be instantiated just like an abstract class.

A Java Interface is also a virtual construct in the programming world but is contrary to the Java Class, Interface cannot be instantiated or created as an object. An Interface denotes a group of logical entities. It can also act as a contract between two subsystems while communicating with each other.

(b) $PQ \cdot R / ST++$

(c) Row Major Address Formula: $M[i][j] = B(A) + W[(i - I_r) \cdot \text{column} + (j - I_c)]$

In the given problem, Address of $P[10][7]$ is asked.

1400 is base address, column = 10, $W = 8$ bytes, $i = 10$, $j = 7$, $I_r = 0$ and $I_c = 0$.

Address of $P[10][7]$:

$$\begin{aligned}
 &= 1400 + 8[(10 - 0) \cdot 10 + (7 - 0)] \\
 &= 1400 + 8[100 + 7] \\
 &= 1400 + 8 \cdot 107 \\
 &= 2256
 \end{aligned}$$

(d) (i) $O(a + bc)$

(ii) $O(n^2)$

(e)

| Constructor | Method |
|--|---|
| 1. A constructor is used to initialize the state of an object. | 1. The method is used to expose the behaviour of an object. |

| | |
|---|--|
| 2. The constructor must not have a return type. | 2. The method must have a return type. |
| 3. Constructor name must be same as the class name. | 3. Method name may or may not be the same as a class name. |

Question 3.

The following function magicfun() is a part of some class. What will the function magicfun() return, when the value of n=7 and n=10, respectively? Show the dry run/working: [5]

```
int magicfun (int n)
{
if(n == 0)
return 0;
else
return magicfun(n/2) * 10 + (n%2);
}
```

Answer:

At n = 7 ⇒ 111

At n = 10 ⇒ 1010

Binary equivalent of N

Part – II (50 Marks)

Answer six questions in this part, choosing two questions from Section -A, two from Section – B and two from Section – C.

Section – A

Answer any two questions.

Question 4.

(a) Given the Boolean function $F(A, B, C, D) = \Sigma(2, 3, 4, 5, 6, 7, 8, 10, 11)$.

(i) Reduce the above expression by using 4-variable Karnaugh map, showing the various groups (i.e. octal, quads and pairs). [4]

(ii) Draw the logic gate diagram for the reduced expression. Assume that the variables and their complements are available as inputs. [1]

(b) Given the Boolean function $F(P, Q, R, S) = \Sigma(0, 1, 2, 4, 5, 6, 8, 10)$.

(i) Reduce the above expression by using the 4-variable Karnaugh map, showing the various groups (i.e., octal, quads and pairs). [4]

(ii) Draw the logic gate diagram for the reduced expression. Assume that the variables and their complements are available as inputs. [1]

Answer:

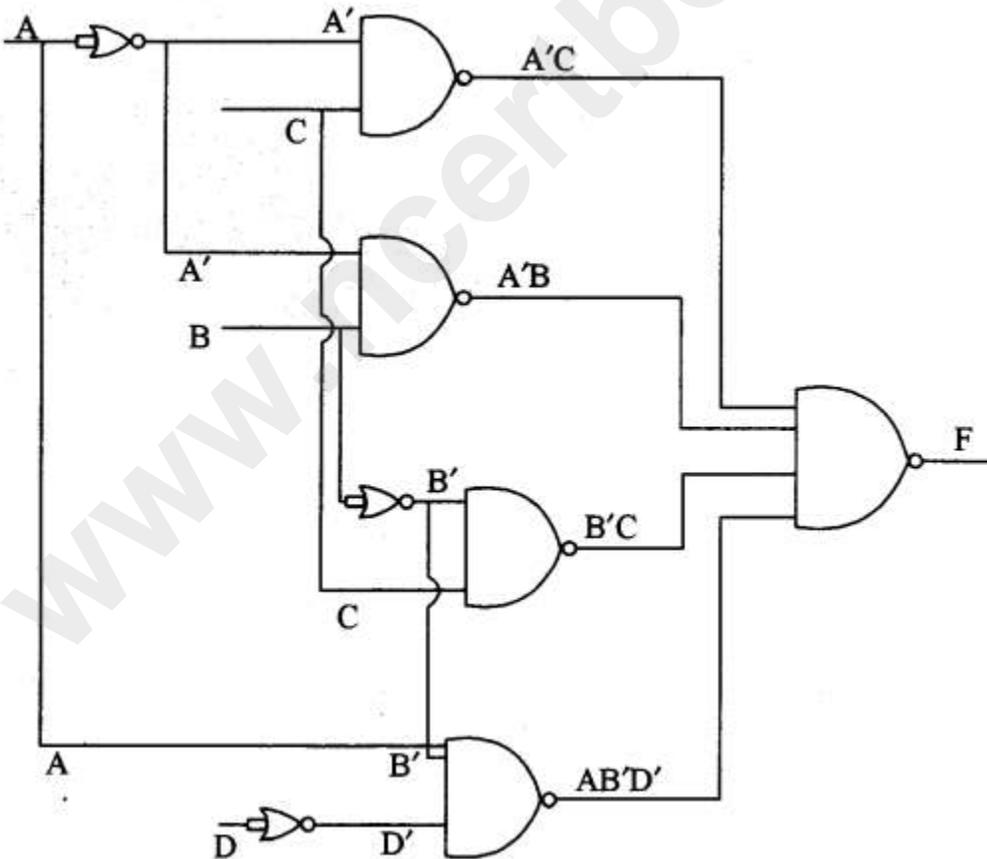
(a) (i) $F(A, B, C, D) = \Sigma (2, 3, 4, 5, 6, 7, 8, 10, 11)$

| | CD | C'D' | C'D | CD | CD' |
|------|----|------|-----|----|-----|
| A'B | | | | 1 | 1 |
| A'B' | | 0 | 1 | 1 | 2 |
| A'B | 1 | 1 | | 1 | 1 |
| AB | | | | | |
| AB | | 12 | 13 | 15 | 14 |
| AB' | 1 | | | 1 | 1 |
| | | 8 | 9 | 11 | 10 |

$$= CA' + A'B + B'C + AB'D'$$

$$= A'C + A'B + B'C + AB'D'$$

(ii) Logic gate diagram:

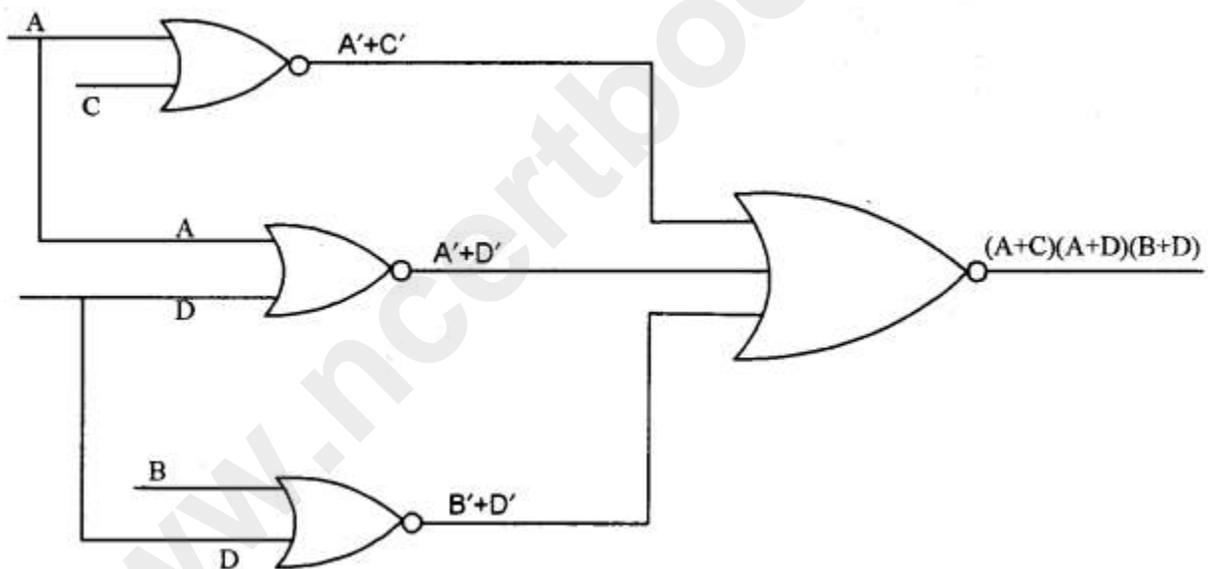


(b) (i) $F(P, Q, R, S) = \pi (0, 1, 2, 4, 5, 6, 8, 10)$

| CD | C+D | C+D' | C'+D' | C'+D |
|-------|--------|--------|-------|---------|
| A+B | 0 0 | 0 1 | 3 | 0 2 |
| A+B' | 0 4 | 0 5 | 7 | 0 6 |
| A'+B' | 12 | 13 | 15 | 14 |
| A'+B | 0 8 | 9 | 11 | 0 10 |

$$= (A + C)(A + D)(B + D)$$

(ii) Logic gate diagram:



Question 5.

(a) A school intends to select candidates for an Inter-School Essay Competition as per the criteria are given below: [5]

The student has participated in an earlier competition and is very creative.

OR

The student is very creative and has excellent general awareness, but has not participated in any competition earlier.

OR

The student has excellent general awareness and has won a prize in an inter-house

competition.
The inputs are:

| | |
|--------|---|
| Inputs | |
| A | participated in a competition earlier |
| B | is very creative |
| C | won a prize in an inter-house competition |
| D | has an excellent general awareness |

(In all the above cases 1 indicates yes and 0 indicates no).

Output: X [1 indicates yes, 0 indicates no for all cases]

Draw the truth table for the inputs and outputs given above and write the POS expression for X(A, B, C, D).

(b) State the application of a Half Adder. Draw the truth table and circuit diagram for a Half Adder. [3]

(c) Convert the following Boolean expression into its canonical POS form: [2]

$$F(A, B, C) = (B + C')(A' + B)$$

Answer:

(a) Truth Table:

| A | B | C | D | X | |
|---|---|---|---|---|-----------------|
| 0 | 0 | 0 | 0 | 0 | m ₀ |
| 0 | 0 | 0 | 1 | 0 | m ₁ |
| 0 | 0 | 1 | 0 | 0 | m ₂ |
| 0 | 0 | 1 | 1 | 1 | m ₃ |
| 0 | 1 | 0 | 0 | 0 | m ₄ |
| 0 | 1 | 0 | 1 | 1 | m ₅ |
| 0 | 1 | 1 | 0 | 0 | m ₆ |
| 0 | 1 | 1 | 1 | 1 | m ₇ |
| 1 | 0 | 0 | 0 | 0 | m ₈ |
| 1 | 0 | 0 | 1 | 0 | m ₉ |
| 1 | 0 | 1 | 0 | 0 | m ₁₀ |
| 1 | 0 | 1 | 1 | 1 | m ₁₁ |
| 1 | 1 | 0 | 0 | 1 | m ₁₂ |
| 1 | 1 | 0 | 1 | 1 | m ₁₃ |
| 1 | 1 | 1 | 0 | 1 | m ₁₄ |
| 1 | 1 | 1 | 1 | 1 | m ₁₅ |

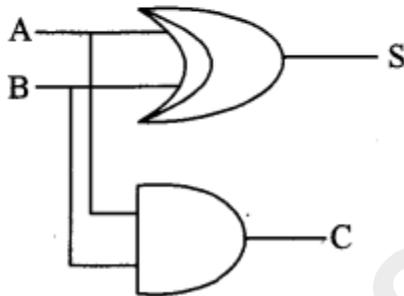
The POS expression is,
 $X(A, B, C, D) = \pi(0, 1, 2, 4, 6, 8, 9, 10)$

(b) Half adder is a combinational logic circuit with two inputs and two outputs. The half adder circuit is designed to add two single bit binary number A and B. It is the basic building block for the addition of two single-bit numbers. This circuit has two outputs carry and sum.

Truth Table:

| Inputs | | Outputs | |
|--------|---|---------|---|
| A | B | S | C |
| 0 | 0 | 0 | 0 |
| 0 | 1 | 1 | 0 |
| 1 | 0 | 1 | 0 |
| 1 | 1 | 0 | 1 |

Circuit Diagram :



(c) $X + YZ = (X + Y)(X + Z)$

Now $A.A' = 0$, Similarly, $C.C' = 0$

Therefore, $(B + C') = (B + C' + A.A') = (B + C + A)(B + C + A')$

[Treating $B + C'$ as a single variable]

and $(A' + B) = (A' + B + C.C') = (A' + B + C)(A' + B + C')$

[Treating $A' + B$ as a single variable]

$$F(A, B, C) = (B + C').(A' + B)$$

$$= (B + C' + A)(B + C' + A')(A' + B + C)(A' + B + C')$$

Question 6.

(a) What is & Multiplexer? How is it different from a decoder? Draw the circuit diagram for a 8 : 1 Multiplexer. [5]

(b) Prove the Boolean expression using Boolean laws. Also, mention the law used at each step.

$$F = (x' + z) + [(y' + z).(x' + y)]' = 1 \quad [3]$$

(c) Define maxterms and mm terms. Find the maxterm and mintenn when: [2]

$P = 0, Q = 1, R = 1$ and $S = 0$

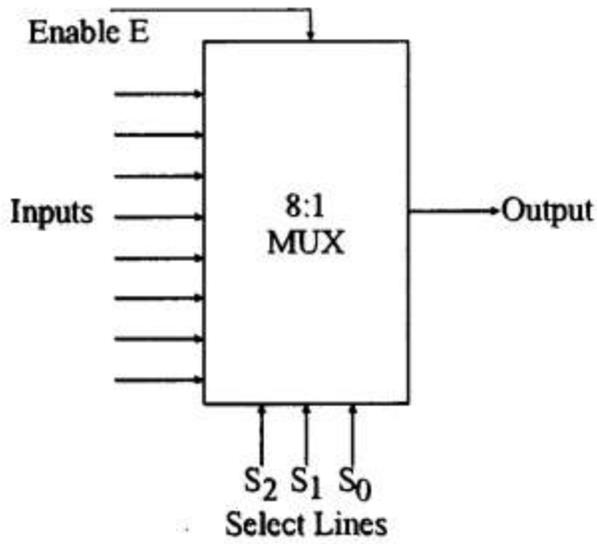
Answer:

(a) Multiplexer is a special type of combinational circuit. There are n -data inputs, one output and m select inputs with $2^m = n$. It is a digital circuit which selects one of the n data inputs and routes it to the output.

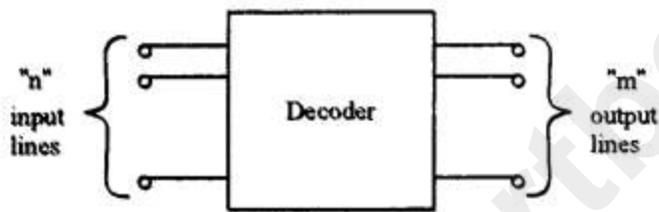
The main function of multiplexer basically is to connect information from one point to another point through wires while on the other hand, decoders convert the outputs for several operations such as data collection and calculations.

Multiplexer:

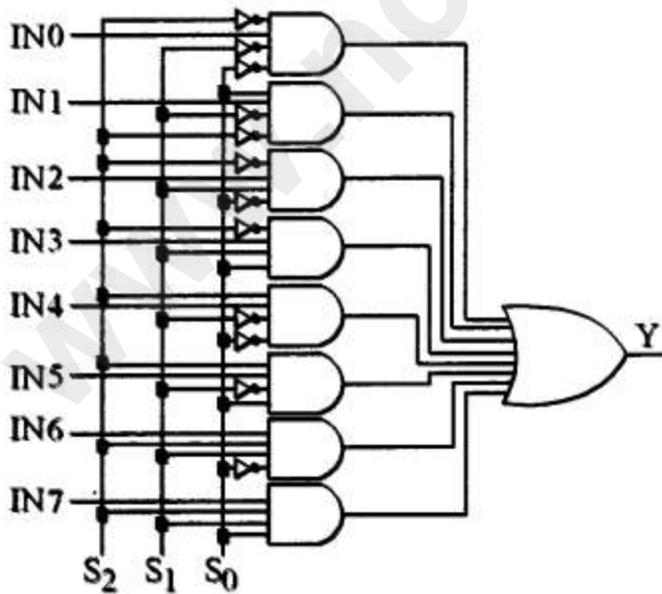
www.ncertbooks.net



Decoder :



Circuit diagram for 8 : 1 MUX :



$$\begin{aligned}
(b) \quad F &= (x' + z) + [(y' + z).(x' + y)]' \\
&= x' + z + (y' + z)' + (x' + y)' \\
&= x' + z + (y')' z' + (x')' y' \text{ [De Morgan's Law]} \\
&= x' + z + yz' + xy' \text{ [Double Negation]} \\
&= x' + xy' + z + yz' \text{ [} a + a'b = a + b \text{]} \\
&= x' + y' + z + y \\
&= x' + z + y' + y \text{ [Complement Law: } a + a' = 1 \text{]} \\
&= x' + z + 1 \\
&= 1
\end{aligned}$$

(c) A minterm is a product (AND) of all variables taken either in their direct or complemented form. Any Boolean function can be expressed as a sum of its 1-minterms and the inverse of the function can be expressed as a sum of its 0-minterms. Therefore,

$F(\text{list of variables}) = \Sigma (\text{list of 1-minterm indices})$
and $F'(\text{list of variables}) = \Sigma (\text{list of 0-minterm indices})$

General example: $F = \Sigma(1, 2, 4, 7)$

A maxterm is an addition (OR) of all variables taken either in their direct or complemented form. Any Boolean function can be expressed as a product of its 0-max terms and the inverse of the function can be expressed as a product of its 1-max terms. Therefore, $F(\text{list of variables}) = \pi (\text{list of 0-maxterm indices})$

and $F'(\text{list of variables}) = \pi (\text{list of 1-maxterm indices})$

General example:

$$F = \pi(0, 3, 5, 6)$$

Here, we are given,

$$P = 0, Q = 1, R = 1 \text{ and } S = 0$$

$$\Rightarrow \text{Minterm} = P'QRS'$$

$$\Rightarrow \text{Maxterm} = (P + Q' + R' + S)$$

Section – B

Answer any two questions.

- Each program should be written in such a way that it clearly depicts the logic of the problem.
- This can be achieved by using mnemonic names and comments in the program.
- Flowcharts and Algorithms are not required.
- The programs must be written in Java.

Question 7.

A class Palin has been defined to check whether a positive number is a Palindrome number or not. [10]

The number 'N' is palindrome if the original number and it's reverse are the same.

Some of the members of the class are given below:

Class name: Palin

Data members/instance variables:

num: integer to store the number

revnum: integer to store the reverse of the number

Methods/Member functions:

Palin(): constructor to initialize data members with legal initial values

void accept(): to accept the number

int reverse(int y): reverses the parameterized argument 'y' and stores it in revenue using a recursive technique

void check(): checks whether the number is a Palindrome by invoking the function reverse() and display the result with an appropriate message

Specify the class Palin giving the details of the constructor (), void accept(), int reverse(int) and void check(). Define the main() function to create an object and call the functions accordingly to enable the task.

Answer:

```
import java.io.*;
class Palin {
    int num;
    int revnum;
    Palin() {
        num = 0;
        revnum = 0;
    }
    void accept() throws IOException {
        BufferedReader y = new BufferedReader(new
        InputStreamReader(System.in));
        System.out.print("Enter the Number:");
        String a = y. readLine();
        num = Integer.parseInt(a);
    }
    int reverse(int y) {
        int len = (y + " ").length();
        if(len ==1) {
            return y;
        }
        else
        {
            return(((y%10)* (int) Math.powf 10. len - 1)) + reverse(y /10));
        }
    }
    void check(){
        revnum = reverse(num);
        if(num == revnum) {
            System.out.println("\n Nrrnber is palindrome");
        }
    }
}
```

```

else {
System.out.println("\n Number is not palindrome");
}
}
public static void main(String args[]) throws IOException {
    Palin p = new Palin();
    p.accept();
    p.check();
}
}

```

Question 8.

A class Adder has been defined to add any two accepted time. [10]

Example:

Time A – 6 hours 35 minutes

Time B – 7 hours 45 minutes

Their sum is – 14 hours 20 minutes (where 60 minutes = 1 hour)

The details of the members of the class are given below:

Class name: Adder

Data member/instance variable:

a[]: integer array to hold two elements (hours and minutes)

Member functions/methods:

Adder (): constructor to assign 0 to the array elements

void readtime (): to enter the elements of the array

void addtime (Adder X, Adder Y): adds the time of the two parameterized objects X and Y and stores the sum in the current calling object

void disptime(): displays the array elements with an appropriate message (i.e., hours= and minutes=)

Specify the class Adder giving details of the constructor(), void readtime(), void addtime(Adder, Adder) and void disptime(). Define the main() function to create objects and call the functions accordingly to enable the task.

Answer:

```

import java.io.*;
class Adder {
    inta[];
    Adder() {
        a = new int[2];
    }
    void readtime() throws IOException {
        InputStreamReader x = new InputStreamReader(System.in);
        BufferedReader y = new BufferedReader(x);
        System.out.println(":: Time");
        System.out.println("Enter hour:");
    }
}

```

```

a[0] = Integer.parseInt(y.readLine());
System.out.println("Enter minute :");
a[1] = Integer.parseInt(y.readLine());
}
void addtime(Adder X, Adder Y) {
int hour1 = X.a[0];
int min1 = X.a[1];
int hour2 = Y.a[0];
int min2 = Y.a[1];
int hourSum = hour1 + hour2;
int minSum = min1 + min2;
a[0] = hourSum + (minSum/60);
a[1] = minSum%60;
}
void disptime() {
System.out.println("Their sum is-");
System.out.println("hours =" + a[0] + " minutes =" + a[1]);
}
public static void main(String args[ ]) throws IOException {
Adder obj 1 = new Adder();
Adder obj2 = new Adder();
Adder sumObj = new Adder();
obj1.readtime();
obj2.readtime();
sumObj.addtime(obj1, obj2);
sumObj.disptime();
}

```

Question 9.

A class SwapSort has been defined to perform string related operations on a word input. Some of the members of the class are as follows: [10]

Class name: SwapSort

Data members/instance variables:

wrd: to store a word

len: integer to store the length of the word

swapwrd: to store the swapped word

sortwrd: to store the sorted word

Member functions/methods:

SwapSort(): default constructor to initialize data members with legal initial values

void readword(): to accept a word in UPPER CASE

void swapchar(): to interchange/swap the first and last characters of the word in 'wrd' and stores the new word in 'swapwrd'

void sortword(): sorts the characters of the original word in alphabetical order and stores it in 'sortwrd'

void display(): displays the original word, swapped word and the sorted word

Specify the class SwapSort, giving the details of the constructor(), void readword(), void

swapchar(), void sortword() and void display(). Define the main() function to create an object and call the functions accordingly to enable the task.

Answer:

```
import java.io.*;
public class SwapSort {
    String wrd;
    int len;
    String swapwrđ;
    String sortwrđ;
    SwapSort() {
        wrđ = " ";
        len = 0;
        swapwrđ = " ";
        sortwrđ = " ";
    }
    void readword() throws IOException {
        InputStreamReader x = new InputStreamReader(System.in);
        BufferedReader y = new BufferedReader(x);
        System.out.println("Enter word:");
        wrđ = y.readLine().toUpperCase();
    }
    void swapwrđ() {
        String w = wrđ;
        swapwrđ = w.charAt(w.length() - 1) + w.substring(1, w.length()
        - 1) + w.charAt(0);
    }
    void sortwrđ () {
        String w = wrđ;
        char[] charArray = w.toCharArray();
        int length = charArray.length;
        for(int i = 0; i < length; i++) {
            for(int j = i + 1; j < length; j++) {
                if(charArray[j] < charArray[i]) {
                    char temp = charArray[i];
                    charArray[i] = charArray[j];
                    charArray[j] = temp;
                }
            }
        }
        for(char c : charArray) {
            sortwrđ = sortwrđ + c;
        }
    }
    void display() {
```

```

System.out.println("Original word : " + wrd);
System.out.println("Swapped word : " + swapwrđ);
System.out.println("Sorted word : " + sortwrđ);
}
public static void main(String args[ ]) throws IOException {
SwapSort obj = new SwapSort();
obj.readword();
obj.swapwrđ();
obj.sortwrđ();
obj.display!);
}
}

```

Section – C

Answer any two questions.

- Each program should be written in such a way that it clearly depicts the logic of the problem stepwise.
- This can be achieved by using comments in the program and mnemonic names or pseudo-codes for algorithms.
- The programs must be written in Java and the algorithms must be written in general/standard form, wherever required/specified.
- Flowcharts are not required.

Question 10.

A superclass Product has been defined to store the details of a product sold by a wholesaler to a retailer. Define a subclass Sales to compute the total amount paid by the retailer with or without fine along with service tax. [5]

Some of the members of both classes are given below:

Class name: Product

Data members/instance variables:

name: stores the name of the product

code: integer to store the product code

amount: stores the total sale amount of the product (in decimals)

Member functions/methods:

Product (String n, int c, double p): parameterized constructor to assign data members:

name = n, code = c and amount = p

void show(): displays the details of the data members

Class name: Sales

Data members/instance variables:

day: stores number of days taken to pay the sale amount

tax: to store the service tax (in decimals)

totalamt: to store the total amount (in decimals)

Member functions/methods:

Sales(...): parameterized constructor to assign values to data members of both the classes

void compute(): calculates the service tax @ 12.4% of the actual sale amount
calculates the fine @ 2.5% of the actual sale amount only if the amount paid by the retailer to the wholesaler exceeds 30 days calculates the total amount paid by the retailer as (actual sale amount + service tax + fine)
void show (): displays the data members of the superclass and the total amount
Assume that the superclass Product has been defined. Using the concept of inheritance, specify the class Sales giving the details of the constructor (...), void compute()) and void show(). The superclass, main function and algorithm need NOT be written.

Answer:

```
import java.io.*;
class Product
{
String name;
int code;
double amount;
Product(String n, int c, double p) {
name = n;
code = c;
amount = p;
}
void show() {
System.out.println("Name is :"+ name);
System.out.println("Code is : " + code);
System.out.println("Total Sale Amount:" + amount);
}
}
class Sales extends Product {
int day;
double tax;
double totamt;
double fine = 0.0;
Sales(String n, int c, double p, int d) {
super(n, c, p);
day = d;
}
void compute() {
if(day < 30){ tax = 12.4 * amount /100; totamt = amount + tax; }
if(day > 30) {
tax= 12.4 * amount /100;
fine = 2.5 * amount /100;
totamt = amount + tax + fine;
}
}
}
```

```

void show () {
show();
System.out.println("Total amount to be paid::"+ totamt);
}
}

```

Question 11.

A queue is an entity which can hold a maximum of 100 integers. The queue enables the user to add integers from the rear and remove integers from the front. [5]

Define a class Queue with the following details:

Class name: Queue

Data members/instance variables:

Que[]: array to hold the integer elements

size: stores the size of the array

front: to point the index of the front

rear: to point the index of the rear

Member functions:

Queue(int mm): constructor to initialize the data

size = mm, front = 0, rear = 0

void addele(int v): to add integer from the rear if possible else display the message "Overflow"

int delele(): returns elements from front if present, otherwise displays the message "Underflow" and return-9999

void display(): displays the array elements

Specify the class Queue giving details of ONLY the functions void addele(int) and int delete()

Assume that the other functions have been defined.

The main function and algorithm need NOT be written.

Answer:

```

void addele(int v) {
iffrear == Que.length-1) {
System.out.println("Overflow");
}
else {
Que[rear ++] = v;
}}
int delele() {
if(front == 0){
System.out.println("Underflow");
return-9999;
else {
return Que[front--];
}
}

```

}

Question 12.

(a) A linked list is formed from the objects of the class Node. The class structure of the Node is given below: [2]

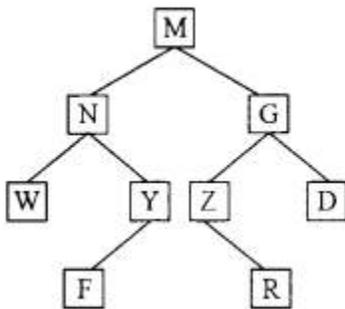
```
class Node
{
int num;
Node next;
}
```

Write an Algorithm OR a Method to count the nodes that contain only odd integers from an existing linked list and returns the count.

The method declaration is as follows :

int CountOdd (Node startPtr)

(b) Answer the following questions from the diagram of a Binary Tree given below:



(i) Write the postorder traversal of the above tree structure. [1]

(ii) State the level numbers of the nodes N and R if the root is at 0 (zero) level. [1]

(iii) List the internal nodes of the right sub-tree. [1]

Answer:

```
(a) int countOddNodes(Node myNode) {
int count = 0
while(myNode.next != null) {
if(myNode.num%2 == 0) {
count ++;
}
}
myNode= myNode.next;
}
return count;
}
```

(b) (i) Postorder traversal: W → F → Y → N → R → Z → D → G → M

(ii) N is at level 1 and R is at level 3

(iii) Internal nodes in right subtree = G, Z.