

## B. ANALYSIS OF PERFORMANCE

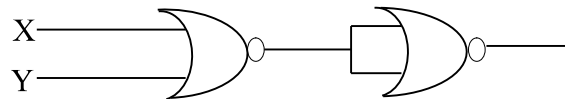
### PART I

Answer all questions.

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

#### Question 1

- (a) Simplify:  $(A + C) \cdot (A + A \cdot D) + A \cdot C + C$  [2]
- (b) Draw a logic circuit for  $(A + B) \cdot (C + D) \cdot C$  [2]
- (c) Verify the following proposition with the help of a truth table: [2]  
 $P \vee (\sim P \wedge Q) = P \vee Q$
- (d) State De Morgan's law and verify it, using a truth table. [2]
- (e) Answer the questions related to the circuit given below: [2]



- (i) Give the output if,  $X=1$  and  $Y=0$
- (ii) Name the basic gate represented by the above diagram.

#### Comments of Examiners

- (a) Most of the candidates answered well. Some reduced the expression in one step without showing the working. In a few cases, the laws were used incorrectly. Some candidates were unable to open the brackets in the expression
- (b) Some candidates interchanged the gates. i.e. OR gate with AND gate and vice versa.
- (c) This part was well answered by most candidates. Some were confused with symbols ' $\wedge$ ' and ' $\vee$ ' and interchanged them in the truth table. Some solved algebraically instead of using the truth table.
- (d) A number of candidates proved both the laws of De Morgan. A few candidates mentioned some other law. Some wrote 'Break the line and change the sign' which is not the law, but a way to remember the law.
- (e) (i) Most of the candidates attempted this part well except for a few who gave the answer as 0.
- (ii) A few candidates mentioned the name of the gate used in the circuit instead of the basic gate the circuit represents. i.e NOR gate instead of OR gate.

#### Suggestions for teachers

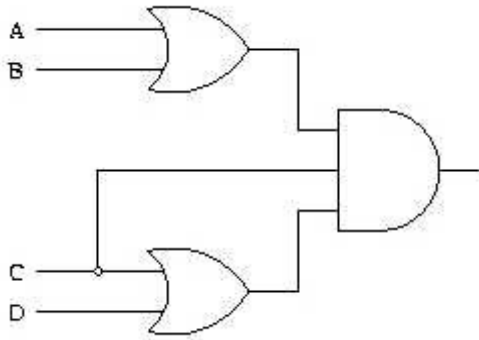
- Students should be told to show the working and mention the laws, if necessary in such type of questions. Knowledge of the laws and practice in their application is important.
- Logic gates and logic circuits must be practiced with almost every expression.
- Propositional logic should be taught using all terms that are required. The symbols used in propositions must be explained.
- All the laws of Boolean algebra must be practiced and proved with the help of truth table.
- Logic circuits with objective type expression must be done.
- More practice must be given for Universal Gates.

**MARKING SCHEME**

**Question 1.**

- (a) Simplify:  $(A + C)(A + AD) + AC + C$   
 $(A + C)A + AC + C$   
 $A + AC + AC + C$   
 $A + C$

- (b) **Logic circuit for  $(A+B) \cdot (C+D) \cdot C$**



- (c) Proving of  $P \vee \sim P \wedge Q = P \vee Q$

				LHS	RHS
P	Q	$\sim P$	$\sim P \wedge Q$	$P \vee \sim P \wedge Q$	$P \vee Q$
0	0	1	0	0	0
0	1	1	1	1	1
1	0	0	0	1	1
1	1	0	0	1	1

- (d) **De Morgan's law** states that the complement of sum of the variables is same as product of the individual complements of the variables and vice-versa.

$(A.B)' = A' + B'$        $(A+B)' = A' \cdot B'$

A	B	A.B	$(A.B)'$	$A'+B'$
0	0	0	1	1
0	1	0	1	1
1	0	0	1	1
1	1	1	0	0

- (e) (i) 1  
(ii) OR gate

## Question 2

- (a) Define *computational complexity*. Calculate the complexity using Big‘O’ notation [2]  
for the following code segment:  
for(int k=0;k<n;k++)  
    s+=k;
- (b) Convert the following infix notation into postfix form: [2]  
 $X + ( Y - Z ) + (( W + E ) * F ) / J$
- (c) Differentiate between *this keyword* and *super key word*. [2]
- (d) The array D[-2...10][3...8] contains double type elements. If the base address is [2]  
4110, find the address of D[4][5], when the array is stored in **Column Major Wise**.
- (e) State *any two* characteristics of a Binary tree. [2]

### Comments of Examiners

- (a) This part was answered well by most of the candidates. Several candidates used examples to illustrate the answer. A few candidates found it difficult to provide appropriate definition. Some candidates wrote ‘k’ instead of ‘n’ for complexity
- (b) Most candidates were able to solve this problem correctly. Some candidates wrote the correct answer without showing the working. Some applied the postfix correctly, but could not derive the final answer. Order of precedence was not clear and the placement of ‘+’ sign was improper in several cases.
- (c) Examples were used by a number of candidates to show the difference. In some cases, only ‘super’ keyword was explained.
- (d) A number of candidates were confused with the width of the cell and assumed different values. Some wrote the answer directly without showing the working or mentioning the formula.
- (e) Various answers were given by candidates. Some explained with the help of a diagram.

### Suggestions for teachers

- More practice must be given in complexity of a code or a segment. All definitions related to complexity must be covered.
- Examples need to be practiced with conversion of Infix to Postfix notation using direct method or the stack method. The order of precedence must be explained.
- Java keywords must be explained and practiced in programs for better understanding.
- Size of various data types must be explained as it is necessary in such type of questions. Formula and working should also be shown.
- More practice must be given in Binary Tree, its parts and characteristics.

## MARKING SCHEME

### Question 2.

- (a) Computational complexity is the growth rate or measurement of an algorithm. The level in difficulty in solving mathematically posed problems as measured by the time, number of steps or arithmetic operations, or memory space required (called time complexity, computational complexity, and space complexity, respectively).

Complexity of the segment = O(n)

- (b) =  $X + ( Y - Z ) + ( ( W + E ) * F ) / J$

$$= X + Y Z - + ( W E + * F ) / J$$

$$= X + Y Z - + W E + F * / J$$

$$= X + Y Z - + W E + F * J /$$

$$= X Y Z - + W E + F * J / +$$

(c) this ( ) is used to refer to the instance variables of the class.

super ( ) is used to invoke the super class's constructor/ instance variable/methods

(d) The given values are:

$$B = 4110, W = \mathbf{8 \text{ bytes or } 64 \text{ bits}}$$
 for double type ,  $I = 4, J = 5, Lr = -2, Lc = 3,$

$$M = (10 - (-2)) + 1 = 10 + 2 + 1 = 13$$

$$\text{Address of } A [ I ][ J ] = B + W * [ ( I - Lr ) + M * ( J - Lc ) ]$$

$$= 4110 + 8 * [ (4 - (-2)) + 13 * (5 - 3) ]$$

$$= 4110 + 8 * [6 + 13 * 2]$$

$$= 4110 + 8 * [6 + 26]$$

$$= 4110 + 8 * [32] = 4110 + 256$$

$$= 4366 \text{ (Ans)}$$

$$\text{OR } = 4110 + 64 * [32] = 4110 + 2048$$

$$= 6158 \text{ (Ans)}$$

(e) 1. It is acyclic

2. No two node can be similar or identical

3. It is recursive

4. There is only one unique path between two nodes.

5. Rooted data structure OR Linear data structure OR Dynamic data structure

6. Every node has a maximum of two sub nodes ( or maximum degree 2 )

### Question 3

(a) The following function is a part of some class. Assume 'x' and 'y' are positive integers, greater than 0. Answer the given questions along with dry run / working.

```
void someFun(int x, int y)
{
    if(x>1)
    {
        if(x%y==0)
        {
            System.out.print(y+" ");
            someFun(x/y, y);
        }
        else
            someFun(x, y+1);
    }
}
```

- (i) What will be returned by `someFun(24,2)`? [2]
  - (ii) What will be returned by `someFun(84,2)`? [2]
  - (iii) State in one line what does the function `someFun()` do, apart from recursion? [1]
- (b) The following is a function of some class which checks if a positive integer is an Armstrong number by returning true or false. (*A number is said to be Armstrong if the sum of the cubes of all its digits is equal to the original number.*) The function does not use modulus (%) operator to extract digit. There are some places in the code marked by ?1?, ?2?, ?3?, ?4?, ?5? which may be replaced by a statement/expression so that the function works properly.

```

boolean ArmstrongNum( int N )
{
    int sum= ?1?;
    int num=N;
    while( num>0)
    {
        int f= num/10;
        int s = ?2?;
        int digit = num- s;
        sum+= ?3?;
        num = ?4?;
    }
    if(?5? )
        return true;
    else
        return false;
}

```

- (i) What is the statement or expression at ?1? [1]
- (ii) What is the statement or expression at ?2? [1]
- (iii) What is the statement or expression at ?3? [1]
- (iv) What is the statement or expression at ?4? [1]
- (v) What is the statement or expression at ?5? [1]

### Comments of Examiners

- (a) Some candidates used examples to illustrate the answer. A number of candidates found it difficult to provide an appropriate definition. A few candidates wrote 'k' instead of 'n' for complexity
- (b) Most candidates were able to solve this problem correctly. Some candidates wrote the correct answer without showing the working. Some applied the postfix correctly, but could not derive the final answer. The order of precedence was not clear and the placement of '+' sign was improper in several cases. A few candidates interpreted the expression as 'sum == num' instead of 'sum == N'.

### Suggestions for teachers

- Students should be asked to show the working and explain with memory blocks. More practice must be given in Recursive techniques.
- Practice should be given to understand the program line by line and derive at a conclusion of what the function is doing.
- More practice should be given on programs using conditions / looping and other output related programs. Teachers are expected to show the dry run/ working of program and emphasize that working is necessary to get full credit. More practice must be given in standard programs.

### **MARKING SCHEME**

#### **Question 3.**

- (a) (i) someFun(24, 2)  
24>1 true      24%2==0 true      output 2  
someFun(24/2, 2)  
12>1 true      12%2==0 true      output 2  
someFun(12/2, 2)  
6>1 true      6%2==0 true      output 2  
someFun(6/2, 2)  
3>1 true      3%2==0 false      call someFun(3, 3)  
someFun(3, 3)  
3>1 true      3%3==0 true      output 3  
someFun(3/3, 3)      1>1 false

#### **OUTPUT: 2 2 2 3**

- (ii) someFun(84, 2)  
84>1 true      84%2==0 true      output 2  
someFun(42/2, 2)  
42>1 true      42%2==0 true      output 2  
someFun(42/2, 2)  
21>1 true      21%2==0 false      call someFun(21, 3)  
someFun(21, 3)  
21>1 true      21%3==0 true      output 3  
someFun(7/3, 3)

7>1 true	7%3==0 false	call someFun(7, 4)
someFun(7, 4)		
7>1 true	7%4==0 false	call someFun(7, 5)
someFun(7, 5)		
7>1 true	7%5==0 false	call someFun(7, 6)
someFun(7, 6)		
7>1 true	7%6==0 false	call someFun(7, 7)
someFun(7, 7)		
7>1 true	7%7==0 true	output 7
someFun(7/7, 7)	1>1 false	

**OUTPUT: 2 2 3 7**

**(iii) Generating Prime factors.**

- (b)
- (i) 0
  - (ii) f \* 10;
  - (iii) digit\*digit\*digit; OR Math.pow(digit,3) ;
  - (iv) num / 10; OR f;
  - (v) sum == N OR N== sum

## PART – II

*Answer **seven** questions in this part, choosing **three** questions from Section A, **two** from Section B and **two** from Section C.*

### SECTION - A

*Answer any **three** questions.*

#### Question 4

- (a) Given the Boolean function  $F(A, B, C, D) = (0,1,2,3,5,7,8,9,10,11)$ .
- (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:
- $$P(A, B, C, D) = ABC'D' + A'BC'D' + A'BC'D + ABC'D + A'BCD + ABCD$$
- (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]

### Comments of Examiners

- (a) (i) Most candidates fared well in this part. Some candidates were not able to draw the K-Map for the POS expression correctly. For a number of candidates the “Map rolling” concept was not very clear. Grey coding and labelling the K-Map was not clear to a number of candidates.
- (ii) Some candidates drew the logic circuit using NOR gates while some others drew vague diagrams.
- (b) (i) Many candidates made errors in place value and putting variables in K-Map. In other cases, the groups were reduced by laws. Several candidates drew the K-Map incorrectly. Some converted the canonical form to cardinal form and then reduced it. Many candidates included the redundant group in the final expression
- (ii) Some candidates drew the logic circuit using NAND gates while some others drew vague diagrams.

### Suggestions for teachers

- Make students reduce POS and SOP expressions using K-Map simultaneously. Students should be told not to include the redundant group in the final expression. Reducing the groups by laws is not needed. Only direct answer for the groups to be written in the reduced expression.
- More and more practice should be given in drawing logic circuits using basic gates and also with universal gates.
- Emphasize on arranging the variables in proper order and the importance of cell values corresponding with the variables. Explain clearly how the group are framed and reduced. Redundant groups are not to be included in the final reduced expression.
- More practice should be given in drawing logic circuits using basic gates and also with universal gates.



**MARKING SCHEME**

**Question 4.**

(a)  $F(A,B,C,D) = \pi(0, 1, 2, 3, 5, 7, 8, 9, 10, 11)$

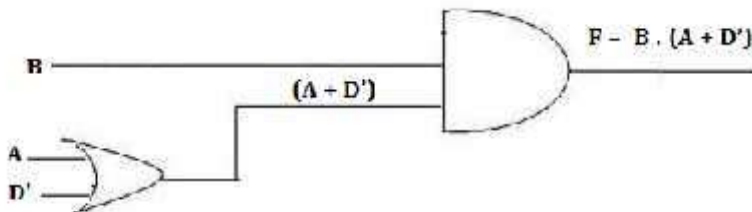
		C+D	C+D'	C'+D'	C'+D
A+B	0	0	0	0	0
A+B'	4	1	0	0	1
A'+B'	12	1	1	1	1
A'+B	8	0	0	0	0

There is one Octal and one Quad :

Octal :  $(M_0 M_1 M_2 M_3 M_8 M_9 M_{10} M_{11}) = B$

Quad :  $(M_1 M_3 M_5 M_7) = A + D'$

Hence  $F(A,B,C,D) = B \cdot (A + D')$



(b)  $P(A,B,C,D) = ABC'D' + A'BC'D' + A'BC'D + ABC'D + A'BCD + ABCD$

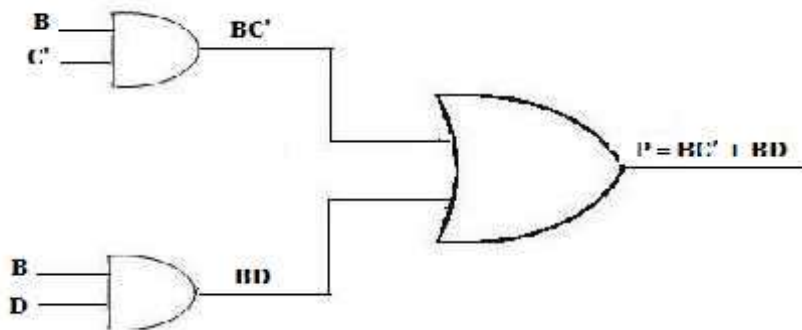
	$C'D'$	$C'D$	$CD$	$CD'$
$A'B'$	0	1	3	2
$A'B$	4	5	7	6
$AB$	12	13	15	14
$AB'$	8	9	11	10

There are two quads :

Quad1 ( $m_4 + m_5 + m_{12} + m_{13}$ ) =  $BC'$

Quad 2 ( $m_5 + m_7 + m_{13} + m_{15}$ ) =  $BD$

Hence  $P(A, B, C, D) = BC' + BD$



### Question 5

A person is allowed to travel in a reserved coach of the train, if he/she satisfies the criteria given below: [10]

- The person has a valid reservation ticket and a valid ID proof.

**OR**

- The person does not have a valid reservation ticket, but holds a valid pass issued by the Railway department with a valid ID proof.

**OR**

- The person is a disabled person and holds a valid pass issued by the Railway department along with a valid ID proof.

The inputs are:

INPUTS	
<b>R</b>	The person has a valid reservation ticket.
<b>P</b>	The person holds a valid pass issued by the Railway department.
<b>D</b>	The person has a valid ID proof.
<b>H</b>	The person is a disabled person.

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

Output : **T** – Denotes allowed to travel (1 indicates yes and 0 indicates no in all the cases)

- (a) Draw the truth table for the inputs and outputs given above and write the **POS** expression for **T(R, P, D, H)**. [5]
- (b) Reduce **T(R, P, D, H)** using Karnaugh map. [5]

Draw the logic gate diagram for the reduced **POS** expression for **T(R, P, D, H)** using only **NOR** gates. You may use gates with two or more inputs. Assume that the variable and their complements are available as inputs.

Comments of Examiners

- (a) A number of candidates were able to attempt this part satisfactorily. Some candidates did not mention the final expression. Some were confused with the POS expression and took the output with 1's instead of 0's. A few candidates took 0's as outputs but wrote the minterms instead of maxterms.
- (b) Many candidates fared well in this part. However, some candidates were not able to draw the K-Map for the POS expression correctly. For a number of candidates the "map rolling" concept was not very clear. Some converted the canonical form to cardinal form and then reduced it. NOR gates in the circuit was ignored by a few candidates.

Suggestions for teachers

- Students should be told to read the question carefully and answer accordingly so that no part is left unanswered. More practice should be given to derive SOP and POS expression from any given truth table (i.e. Minterms and Maxterms)
- Make students reduce POS and SOP expressions using K-Map simultaneously. Students should be told not to include the redundant group in the final expression. The circuit diagram for the reduced expression must be practiced both with basic gates and universal gates.

**MARKING SCHEME**

**Question 5.**

(a)

<b>R</b>	<b>P</b>	<b>D</b>	<b>H</b>	<b>T (OUTPUT)</b>
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	<b>1</b>
0	1	1	1	<b>1</b>
1	0	0	0	0
1	0	0	1	0
1	0	1	0	<b>1</b>
1	0	1	1	<b>1</b>
1	1	0	0	0
1	1	0	1	0
1	1	1	0	<b>1</b>
1	1	1	1	<b>1</b>

**POS Expression:  $T(R, P, D, H) = \pi(0, 1, 2, 3, 4, 5, 8, 9, 12, 13)$**

(b)

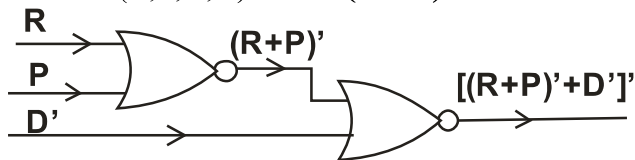
		D+H	D+H'	D'+H'	D'+H
R+P	0	0	0	3	2
R+P'	4	0	0	7	6
R'+P'	12	0	0	15	14
R'+P	8	0	0	11	10

There is one Octal and one Quad :

$$\text{Octal : } (M_0 M_1 M_4 M_5 M_8 M_9 M_{12} M_{13}) = D$$

$$\text{Quad : } (M_0 M_1 M_2 M_3) = R+P$$

Hence  $T(R,P,D,H) = D \cdot (R+P)$



### Question 6

- (a) Draw the truth table and logic gate diagram for an Octal to Binary encoder. [4]
- (b) What is a Multiplexer? State an application of a Multiplexer. Also, draw the logic diagram of a 4:1 Multiplexer. [4]
- (c) Verify the following expression using Boolean laws. Also, mention the law used at each step of simplification. [2]

$$X \cdot Y \cdot Z + X \cdot Y' \cdot Z + X \cdot Y \cdot Z' = X \cdot (Y + Z)$$

Comments of Examiners

- (a) Most of the candidates answered this part correctly. A few candidates drew the decimal encoder while some others drew the gates wrong.
- (b) Some candidates wrote the definition instead of application and in some cases block diagrams were used instead. Some candidates interchanged the gates in the circuit diagram.
- (c) While most candidates answered this part correctly, some did not mentioned the laws. A few candidates wrote the answer directly in one step.

Suggestions for teachers

- More practice should be given in drawing encoder and decoder. Use of proper connector and gates must be explained.
- More practice must be given in drawing of multiplexer. Explanation of various gates and their use must be emphasized
- Tell students that laws must be mentioned while reducing, minimizing or proving an expression.

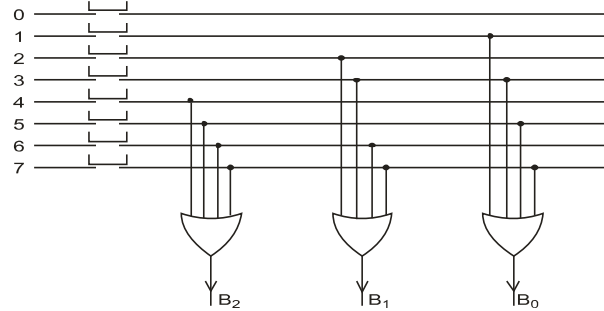
**MARKING SCHEME**

**Question 6.**

(a) Truth table for Octal to Binary Encoder :

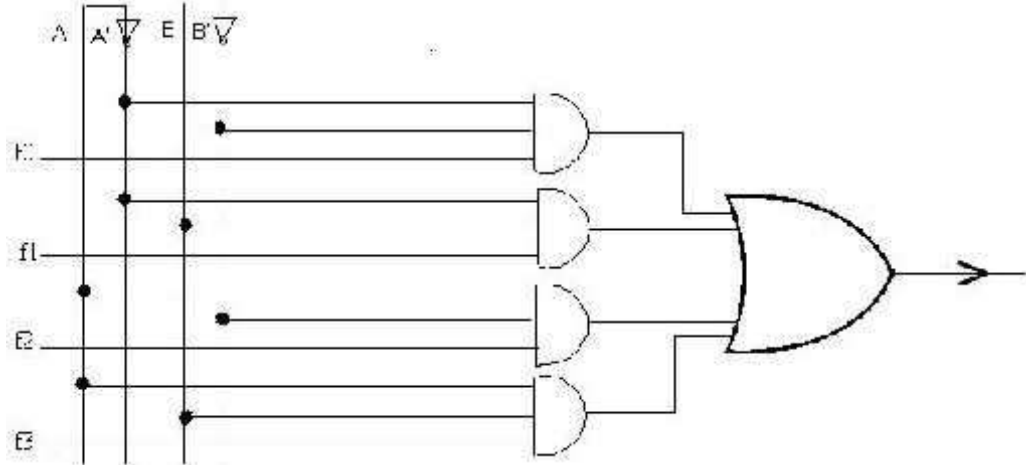
Input	Output		
Octal number	B <sub>2</sub>	B <sub>1</sub>	B <sub>0</sub>
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Logic Diagram for Octal to Binary Encoder :



(b) **Multiplexer:** It is a combinational circuit which inputs parallel data and outputs one serial data.

- Application:**
- (1) It is used as data selector.
  - (2) Routing of signals
  - (3) Data transmission
  - (4) Telephone exchange ,Tv etc.

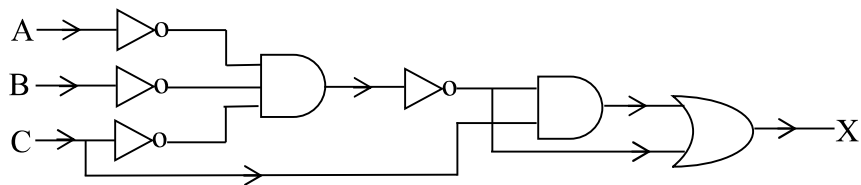


(c)

$$\begin{aligned}
 & X.Y.Z + X.Y'.Z + X.Y.Z' = X.(Y + Z) \\
 & = X.Y.Z + X.Y'.Z + X.Y.Z' \\
 & = X.Y.(Z + Z') + X.Y'.Z && \text{Complimentary law: } Z+Z'=1 \\
 & = X.Y + X.Y'.Z \\
 & = X.(Y + Y'.Z) && \text{Distributive law: } A + B.C = (A+B).(A+C) \\
 & = X.[(Y + Y').(Y+Z)] && \text{Complimentary law: } Y+Y'=1 \\
 & = X.(Y+Z)
 \end{aligned}$$

**Question 7**

- (a) Derive a Boolean expression for the logic circuit given below and reduce the derived expression, using Boolean laws: [3]



- (b) What are universal gates? Construct a logic circuit using NAND gates only for the expression:  $A \cdot (B + C)$  [3]
- (c) Define *Half Adders*. Draw the circuit diagram and the truth table for a Half Adder. [4]

### Comments of Examiners

- (a) Most of the candidates answered this part correctly. Some candidates gave incomplete answers. A few candidates could not reduce the expression. Some wrote the answer directly without mentioning the intermediate points.
- (b) Some candidates used basic gates in drawing the logic circuit instead of NAND gates. The concept of Universal gates not clear to a number of candidates.
- (c) Some candidates drew wrong circuit diagrams. The circuits for partial sum and carry were done separately and not as one circuit in some cases.

### Suggestions for teachers

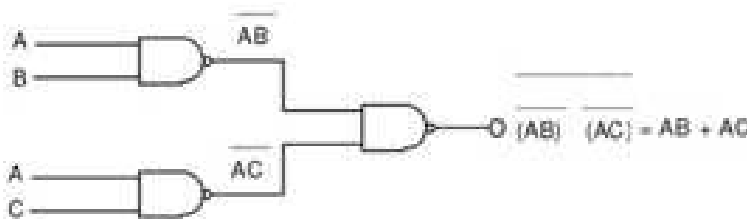
- More practice should be given in deriving expression for any circuit diagram and then reducing it.
- Drawing of logic circuits using universal gates must be explained clearly. Proving of universality of gates must be explained.
- More practice should be given in drawing half adder and full adder and also the truth table and expression of both adders. Students must be told that Half Adder is one circuit and not two different circuits.

### MARKING SCHEME

#### Question 7.

- (a)
  1.  $(A'.B'.C')$
  2.  $(A'.B'.C')'$
  3.  $(A'.B'.C')' . C$
  4.  $(A'.B'.C')' . C + (A'.B'.C')$   
 $(A+B+C) . C + (A+B+C)$   
 $(A+B+C) . (C+1)$   
 $A+B+C$

- (b) Universal gates are derived gates which can perform the function of all basic gates. NAND gate and NOR gate are Universal gates.

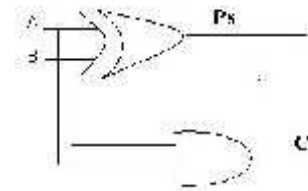
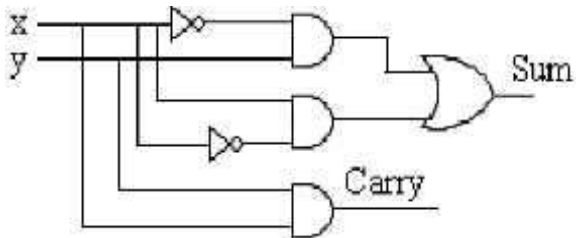


- (c) Half adders are combinational circuits which adds two input binary bits and output two binary bits. (Partial Sum and Carry).



Truth table of half adder :

A	B	Partial Sum	Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1



OR

### 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 8

A class **Admission** contains the admission numbers of 100 students. Some of the data [10] members / member functions are given below:

**Class name** : **Admission**

**Data member/instance variable:**

Adno[ ] : integer array to store admission numbers

**Member functions/methods:**

Admission( ) : constructor to initialize the array elements

void fillArray( ) : to accept the elements of the array in ascending order

int binSearch(int l, int u, int v) : to search for a particular admission number (v) using **binary search** and **recursive technique** and returns 1 if found otherwise returns -1

Specify the class **Admission** giving details of the **constructor**, **void fillArray( )** and **int binSearch(int, int, int)** . Define the **main( )** function to create an object and call the functions accordingly to enable the task.

### Comments of Examiners

Most of the candidates were able to attempt this part correctly. Common errors made by candidates in this question were: the concept of recursion was not clear to some candidates; several candidates did not use the parameters in the function; some took extra variables while others did not initialize the array in the constructor; a few candidates wrote the function binsearch( ) without using the recursive technique. The other function including the constructor was well answered. Some candidates did not write the main( ) function and also ignored the comments to be given in the program.

### Suggestions for teachers

- More practice should be given in solving programs using recursive techniques. Much attention should be paid by the teachers towards recursion and its techniques with examples. Knowledge of base case and recursive case should be given to the students for every program using recursive technique. Invoking function within another function should be given more practice. The students should create recursive code for most iteration cases.

### **MARKING SCHEME**

#### **Question 8.**

```
import java.util.*;
public class Admission
{ static Scanner x=new Scanner(System.in);
  int Adno[]=new int[100];
  Admission(){ }
  void fillArray()
  {
    System.out.println("Enter 100 elements in ascending order ");
    for(int i=0;i<100;i++)
      Adno[i]=x.nextInt();
  }

  int binSearch(int l,int u,int v)
  { int m=(l+u)/2;
    if(Adno[m]==v)
      return l; OR return m;
    else if(l>u)
      return -1;
    else if(Adno[m]>v)
      return binSearch(l,m-1,v);
    else
      return binSearch(m+1,u,v);
  }

  static void main()
  { Admission obj=new Admission();
```

**OR**

```
u = m-1
else
l= m+1
return binSearch (l,u,v);
```

```

obj.fillArray();
System.out.println("Enter value to be searched");
int v=x.nextInt();
int p=obj.binSearch(0,obj.Adno.length-1,v);

if(p==-1)
    System.out.println("No. not found");
else
    System.out.println("No. is present ");
}
}

```

### Question 9

A class **Merger** concatenates two positive integers that are greater than 0 and produces a new merged integer. [10]

**Example:** If the first number is **23** and the second is **764**, then the concatenated number will be **23764**.

Some of the members of the class are given below:

**Class name** : **Merger**

**Data members/instance variables:**

n1 : long integer to store first number  
n2 : long integer to store second number  
mergNum : long integer to store the merged number

**Member functions:**

Merger() : constructor to initialize the data members  
void readNum( ) : to accept the values of the data members n1 and n2  
voidJoinNum( ) : to concatenate the numbers n1 and n2 and store it in mergNum  
void show( ) : to display the original numbers and the merged number with appropriate messages

Specify the class **Merger**, giving the details of the **constructor**, **void readNum( )**, **void JoinNum( )** and **void show( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

### Comments of Examiners

A number of candidates were able to attempt this part well. The joinnum() function was not done properly by some candidates. Various methods/techniques were used to concatenate the numbers. Some did it directly without converting it to String type. Several candidates reversed the number using while loop and then added it to the other number. In some cases 'int' was used instead of 'long' data type. Object creation in the main ( ) function was not properly defined in some cases. Constructor and the main() method was answered properly by a few candidates.

### Suggestions for teachers

- More practice must be given in conversion from numeric to string and string to numeric. That concatenation of strings is possible but does not work on numeric data, must be explained.

### **MARKING SCHEME**

#### **Question 9.**

```
import java.util.*;
public class Merger
{
    long n1,n2,mergNum;
    Merger() { } OR Merger(){ n1= n2 = mergNum = 0;}
    void readNum()
    { Scanner x=new Scanner(System.in);
      System.out.println("Enter two numbers");
      n1=x.nextLong();
      n2=x.nextLong();
    }
    void JoinNum()
    { String s=Long.toString(n1);
      String s1=Long.toString(n2);
      String s2=s+s1;
      mergNum=Long.valueOf(s2);
    }
    void show()
    { System.out.println("First Number= " + n1);
      System.out.println("Second Number= " + n2);
      System.out.println("Merged Number= " + mergNum);
    }
    static void main()
    { Merger obj=new Merger();
      obj.readNum();
    obj.JoinNum();
    obj.show();
  }
}
```

**OR**

```
void JoinNum( )
{ int s=1; long a=n2 ;
  while(a!=0)
  { s=s*10;
    a=a/10;
  }
  mergeNum = (n1 * s) +
n2;
}
```

### Question 10

A class **TheString** accepts a string of a maximum of 100 characters with only one blank space between the words. [10]

Some of the members of the class are as follows:

<b>Class name</b>	:	<b>TheString</b>
<b>Data member/instance variable:</b>		
str	:	to store a string
len	:	integer to store the length of the string
wordcount	:	integer to store the number of words
cons	:	integer to store the number of consonants
<b>Member functions/methods:</b>		
TheString( )	:	default constructor to initialize the data members
TheString( String ds)	:	parameterized constructor to assign str=ds
void countFreq()	:	to count the number of words and the number of consonants and store them in wordcount and cons respectively
void Display( )	:	to display the original string, along with the number of words and the number of consonants

Specify the class **TheString** giving the details of the **constructors**, **void countFreq()** and **void Display( )**. Define the **main( )** function to create an object and call the functions accordingly to enable the task.

#### Comments of Examiners

Different methods / logic was used to extract the words from the sentence. Some candidates used String tokenizer while some others used the split function and charArray to separate words from the sentence. Several candidates did not include the blank space for words. A number of candidates were not able to find the frequency of consonants properly. A few candidates used '|' operator instead of '&&' operator. The main() function, constructor and object creation were not answered by some of the candidates.

#### Suggestions for teachers

- Practice should be given to extract characters from words, words from sentences and sentences from paragraphs. Different methods /logic should be adopted so that wider exposure to string manipulation related programs is given to candidates. Knowledge of constructors to initialize a string and other data members should be given.

## MARKING SCHEME

### Question 10.

```
public class TheString
{
    String str;
    int len,wordcount,cons;
    TheString(){ OR TheString( ) { str=""; wordcount=0; cons=0;}
    TheString(String ds)
    { str=ds; len=str.length();
    }
    void countFreq()
    { char c;
      str=str.toLowerCase ();
      for(int i=0;i<len;i++)
      { c=str.charAt(i);
        if(c==32)
          wordcount++;
        if(c!='a' && c!='e' && c!='i' && c!='o' && c!='u')
          cons++;
      }
      cons = cons – wordcount;
      ++wordcount;
    }
    void Display()
    { System.out.println(" Number of words=" + wordcount ); OR ( wordcount + 1 ) *
      System.out.println(" Number of consonants=" + cons); OR ( cons – wordcount ) *
    }
    static void main()
    { TheString x=new TheString("india is my country");
      x.countFreq();
      x.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 11

**WordPile** is an entity which can hold maximum of 20 characters. The restriction is that a character can be added or removed from one end only.

Some of the members of classes are given below:

**Class name** : **WordPile**

#### **Data members/instance variables:**

ch[ ] : character array to hold the character elements  
capacity : integer variable to store the maximum capacity  
top : to point to the index of the topmost element

#### **Methods/Member functions:**

WordPile( int cap) : constructor to initialise the data member capacity = cap, top = -1 and create the WordPile  
void pushChar( char v) : adds the character to the top of WordPile if possible, otherwise output a message "WordPile is full"  
char popChar() : returns the deleted character from the top of the WordPile if possible, otherwise it returns '\\'

(a) Specify the class **WordPile** giving the details of the **constructor**, **void pushChar(char)** and **char popChar()**. [8]

**The main function and algorithm need not be written.**

(b) What is the name of the entity described above and state *one* of its applications. [2]

### Comments of Examiners

- (a) The concept of stack was not clear to most of the candidates. Common errors made by candidates were: (i) the condition / logic for underflow and overflow was not answered correctly; (ii) increment / decrement of top index was not done properly. The methods pushchar() and popchar() were found to be difficult by some candidates. Some were confused with the Character type array and were not able to return properly. The class declaration and constructors was well answered.
- (b) Some candidates were unable to state the applications and gave vague answers. Some candidates overlooked this part of the question.

### Suggestions for teachers

- More practice should be given in data structure programs like the stacks, queues, de queues, etc. Working must be shown as to how the stack or a queue performs (examples can be supportive). The concept of LIFO and FIFO must be explained to the students with examples from real world. Implementation of stacks, queues and de queues using arrays should be emphasized. Only the concept has to be explained taking the base as an array. It should be made clear to students that it is not an array related program which can be manipulated by shifting / inserting or initializing by any value since these data structures requires pointers and pointers are not supported in java. So, the array is used to show the working of a stack, queue or a de queue
- Stacks and Queues must be explained in details giving all its applications / uses.
- Students should be asked to read the question carefully and answer accordingly.



## MARKING SCHEME

### Question 11.

```
(a) public class WordPile
    {
        char ch[] = new char[20];
        int capacity,top;
        WordPile(int cap)
        { capacity=cap;
          top=-1;
          ch=new char[capacity];
        }
        void pushChar(char v)
        { if(top<capacity-1)
          ch[++top]=v;
          else
          System.out.println("WordPile is full");
        }
        char popChar()
        {
          if(top>=0)
            return ch[top--];
          else
            return '\\';
        }
    }
```

(b) Name of the Entity : **Stack**  
Application : LIFO (Last In First Out )  
Recursion  
Undo  
infix to postfix ( any one)

### Question 12

A line on a plane can be represented by coordinates of the two-endpoints  $p_1$  and  $p_2$  as  $p_1(x_1, y_1)$  and  $p_2(x_2, y_2)$ . [10]

A super class **Plane** is defined to represent a line and a sub class **Circle** to find the length of the radius and the area of circle by using the required data members of super class.

Some of the members of both the classes are given below:

**Class name** : **Plane**

**Data members/instance variables:**

$x_1$  : to store the x-coordinate of the first end point  
 $y_1$  : to store the y-coordinate of the first end point

### Member functions/methods:

Plane( intnx, intny ) : parameterized constructor to assign the data members x1=nx and y1=ny

void Show( ) : to display the coordinates

**Class name** : **Circle**

### Data members/instance variables:

x2 : to store the x-coordinate of the second end point

y2 : to store the y-coordinate of the second end point

radius : double variable to store the radius of the circle

area : double variable to store the area of the circle

### Member functions / methods

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

void findRadius( ) : to calculate the length of radius using the formula:

$$\left( \sqrt{x_2 - x_1^2 + y_2 - y_1^2} \right) / 2$$

assuming that x1, x2, y1, y2 are the coordinates of the two ends of the diameter of a circle

void findArea( ) : to find the area of circle using formula:  $\pi r^2$ . The value of pie ( $\pi$ ) is 22/7 or 3.14

void Show( ) : to display both the coordinates along with the length of the radius and area of the of the circle

Specify the class **Plane** giving details of the **constructor** and **void Show( )**. Using the **concept of inheritance**, specify the class **Circle** giving details of the **constructor**, **void findRadius( )**, **void findArea( )** and **void Show( )**.

**The main function and algorithm need not be written.**

### Comments of Examiners

The concept of inheritance was not clear to a number of candidates. Constructor with inheritance was not answered correctly. Accessing the members of the super class by the derived class, Mathematical functions Math.pow( ) and Math.sqrt( ) were not clear to several candidates. In some cases, double data members were not declared properly. Invoking the show( ) function in the derived class was not answered properly by a few candidates. In some cases, algorithm was written instead of a program. The rest of the function were well answered.

### Suggestions for teachers

- Practice should be given to students on inheritance. Use of constructor using the base class member should be made clear.
- Explain to students the different visibility modes and their accessing capability. Knowledge of calling the member function from the super class to the derived class must be made clear. The keywords 'extends' and 'super' must be given special attention while dealing with inheritance.

## MARKING SCHEME

### Question 12.

```
public class Plane
{
    double x1,y1;
    Plane(double nx, double ny)
    { x1=nx;
      y1=ny;
    }
    void Show()
    { System.out.println("x- coordinate= "+ x1);
      System.out.println("y- coordinate= "+ y1);
    }
}

public class Circle extends Plane
{
    double x2, y2, radius, area;
    Circle(double nx, double ny, double a, double b)
    { super(nx,ny);
      x2=a;
      y2=b;
    }
    void findRadius()
    { radius= ( Math.sqrt( Math.pow((x2-x1),2) + Math.pow((y2-y1),2) ) ) / 2;
    }
    void findArea()
    { area = 3.14 * radius * radius ;
    }
    void Show()
    { super.Show();
      System.out.println("Second x- coordinate= "+ x2);
      System.out.println("Second y- coordinate= "+ y2);
      System.out.println(" Length of radius = "+radius);
      System.out.println(" Area = " + area);
    }
}
```

**Question 13:**

- (a) A linked list is formed from the objects of the class: [4]

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

Write an *Algorithm* **OR** a *Method* to print the sum of nodes that contains only odd integers of an existing linked list.

The method declaration is as follows:

```
void NodesCount( Nodes starPtr )
```

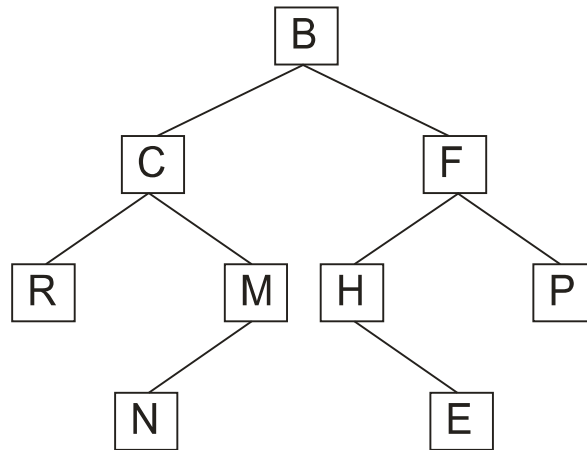
- (b) (i) Give the meaning of the following common expression in Big O notation: [1]

$O(N)$

$O(N^2)$

- (ii) List *any two* cases to analyse algorithm complexities. [1]

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



- (i) Name the leaf nodes of the right sub-tree. [1]

- (ii) Write post order traversal of the left sub-tree of node B including itself. [1]

- (iii) State the level number of nodes R and M when the root is at level 0. [1]

- (iv) Name the internal nodes of the tree. [1]

### Comments of Examiners

- (a) This part was well answered by most candidates. A number of candidates did not create a temporary pointer to the first node. Some candidates had problems in moving the pointer to the next node and checking for null. A few were confused whether to count the nodes or to find the sum of the nodes. Some wrote the algorithm in simple English language covering all the main steps.
- (b) (i) Several candidates were confused and gave vague answers with examples. Some mentioned examples of  $O(n)$  and  $O(n^2)$ .  
(ii) This part was answered well by most of the candidates. Some gave examples to illustrate their answer. A few candidates mentioned the types of complexities instead of cases of complexities.
- (c) (i) Several candidates mentioned the leaf nodes of the entire tree instead of right sub tree.  
(ii) While most candidates answered this part correctly some wrote the post order of the entire tree.  
(iii) This part was well answered by most candidates.  
(iv) Some candidates included root 'B' also in the answer for internal nodes. However, most of the candidates answered this part well.

### Suggestions for teachers

- More programs / algorithms should be practiced with link list and binary tree data structure. Diagrams to illustrate the link list and the Binary Trees must be practiced.
- Definition of complexities / big 'O' and the three cases of complexities must be explained in detail along with examples. The role of the term (N) in complexities must be explained.
- All the three cases of complexities with the factors that influence them must be explained to students. Examples in all three cases must also be given. Also explain the difference between types and cases of complexities to students.
- Explain binary tree with the different parts like root, nodes(internal and external), height, depth, level, size, tree traversal (preorder, inorder and postorder), etc.

### **MARKING SCHEME**

#### **Question 13.**

- (a) **Algorithm to print the sum of the nodes of odd integers in an existing linked list.**

**Steps :**

- 1 - Start
- 2 - Set temporary pointer to start node
- 3 - Repeat steps 4 & 5 until the pointer reaches null. Display the Count. Exit.
- 4 - Check for odd integers and accumulate
- 5 - Move pointer to the next node
- 6 - End

**OR**

**Method to print the sum of the nodes of odd integers in an existing linked list.**

```
void NodeCount( Nodes starPtr)
```

```

        {
        Nodes temp = new Nodes(starPtr)
        int c=0;
        while (temp!=null)
            {
            if( temp.num%2 !=0)
            c=c+temp.num;
            temp=temp.next;
            }

        System.out.println(c);
        }

```

- (b) (i)  $O(n)$  -- It is the complexity of a single loop or condition / expressions  
 $O(n^2)$  – It is a nested loop with each loop going upto n.
- (ii) Best case complexity and Worst case complexity
- (c) i. P , E  
 ii. R , N , M , C , B  
 iii. 2  
 iv. C , M , F , H

**GENERAL COMMENTS:**

**(a) Topics found difficult by candidates in the Question Paper:**

- The symbols ‘ $\wedge$ ’ and ‘ $\vee$ ’ from propositional logic
- Computational complexity definition.
- Size of double data type for the width of an array cell.
- Keywords ‘this’ and ‘super’ with respect to overriding
- Extracting digits without using the % sign.
- K-MAPS (Grouping , map-rolling , place value)
- Deriving POS expression from a truth table
- Recursive technique
- Pre-defined size of an array and re-declaring the size in the constructor.
- Complexity and Big ‘O’ Notation
- Stack operations for adding and removing characters.

**(b) Concepts in which candidates got confused:**

- The symbols in a proposition.
- Difference between ‘this’ and ‘super’.
- Deriving POS expression from Truth table
- Deriving expression from a given circuit and reducing it.
- Concatenation of two numbers.
- Left and right sub tree for a Binary Tree.

**(c) Suggestions for Candidates:**

- Prepare summary for each chapter or use high-lighters to recognize the important terms and definitions.
- Always use latest version for compilers and other software.
- Answers and definitions should be short and precise and according to marks intended. Important words and terms should be underlined or highlighted. Working should be shown at the side of each question where ever required.
- Laws must be mentioned while reducing a Boolean Expression.
- Practice one form of K-Map with proper place value for both SOP and POS.
- In programming documentation is compulsory and should be mentioned with each program.
- Declare the class with data members and member functions. Expand or define each function according to the instructions given by the side of each function.
- Do not memorize the program, try to understand the logic.
- Practice constructors with every program. Treat each function of a class as separate program.