I PRE BOARD EXAMINATION

COMPUTER SCIENCE

Paper 1

(THEORY)

Three 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 II, **choosing two** questions from Section A, **two** questions from Section B, **two** questions from Section C.*

All working including rough work, should be done on the same sheet as the rest of the answers.

The intended marks for questions or parts are given in brackets [ ].

PART - I

Attempt **all** questions.

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

Question 1

(i) The complement of the Boolean expression ~~(X ∘ Y)' + Z'~~ $(X \cdot Y)' + Z'$ is : [1]

(a) $(X + Y) \cdot Z$

(b) $X \cdot Y \cdot Z$

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

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

(ii) Given below are two statements marked Assertion and Reason. Read the two statements carefully and choose the correct option. [1]

Assertion : Recursion utilises more memory as compared to iteration.

Reason : Time complexity of recursion is higher due to the overhead of maintaining the function call stack.

(a) Both Assertion and Reason are true and Reason is the correct explanation for Assertion.

(b) Both Assertion and Reason are true but Reason is not the correct explanation for Assertion.

(c) Assertion is true and Reason is false.

(d) Assertion is false and Reason is true.

(iii) The complement of the reduced expression of $F(A, B) = \Sigma (0, 1, 2, 3)$ is : [1]

(a) 1

(b) $A \cdot B$

(c) 0

(d) $A' + B'$

(iv) The law which represents the Boolean equation $A + B = B + A$ is : [1]

(a) Associative Law

(b) Distributive Law

(c) Commutative Law

(d) Absorption Law

(v) Given below are two statements marked Assertion and Reason. Read the two statements carefully and choose the correct option. [1]

Assertion : The constructor defined in the super class is invoked with the help of super keyword.

Reason : The constructor of super class is not invoked implicitly while creating an object of a sub class.The object of sub class will call the constructor of sub class itself.

(a) Both Assertion and Reason are true and Reason is the correct explanation for Assertion.

(b) Both Assertion and Reason are true but Reason is not the correct explanation for Assertion.

(c) Assertion is true and Reason is false.

(d) Assertion is false and Reason is true.

(vi) If A = 1, B = 0, C = 0 and D = 1, then MAXTERM will be : [1]

(a) A + B + C' + D

(b) AB'C'D

(c) A' + B + C + D'

(d) AB'C'D'

(vii) Name the basic gate that is equivalent to two NAND gates connected in series. [1]

(viii) For Big O notation, state the difference between O(n) and $O(n^2)$. [1]

(ix) A full adder needs five gates and those are 3 AND gates, 1 OR gate and 1 XOR gate. When a full adder is constructed using 2 half adders, it also requires 5 gates. State the names along with the quantity those gates. [1]

(x) Mention one difference between stack and queue. [1]

Question 2

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

(ii) An array ARR[–4 .... 6, –2 .... 12] , stores elements in Row Major Wise, with the address ARR[2][3] as 4142. If each element requires 2 bytes of storage, find the Base address. [2]

(iii) The following function ***getIt( )*** is a part of some class. Assume ***x*** is a positive integer, ***f*** is the lower bound of ***arr[ ]*** and ***l*** is the upper bound of the ***arr[ ]***. Answer the questions given below along with dry run/working.

```
public int getIt(int x,intarr[ ],int f,int l)
    {
        if(f > l)
            return –1;
        int m=(f + l)/2; if(arr[m] < x)
            return getIt(x, m + 1, l);
        else if(arr[m] > x)
            return getIt(x, f, m –1);
        else return m;
    }
```

(a) What will the function ***getIt( )*** return if ***arr[ ] = {10,20,30,40,50}*** and ***x = 40***? [2]

(b) What is function ***getIt( )*** performing apart from recursion? [1]

(iv) The following function is a part of a class which computes the binary equivalent of a decimal number(base 10) by using recursion.There are some places marked as *?1?*, *?2?*, *?3?* which should be replaced by a numerical value/an expression so that the function executes properly.

```
int dec_bin(int n)
{
if(n == 0)
return (?1?);
else
return (dec_bin(n/2) * ?2? + ?3?);
}
```

(a) What is the expression or statement at ?1? [1]

(b) What is the expression or statement at ?2? [1]

(c) What is the expression or statement at ?3? [1]

PART - II [50 MARKS]

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

SECTION - A

*Answer any **two** questions.*

Question 3

(i) Post pandemic, to encourage the tourism industry in India, the Ministry of Tourism started a policy in which a tourist would be allowed to book a resort at a rebate, if the any one of the following criteria matches.

- The tourist has an AADHAR card and has no criminal record.
- The tourist is the government employee and has repeated the resort visit in a span of six months.
- The tourist has an AADHAR card and has repeated the resort visit in a span of six months.

Inputs :

A : The tourist has an AADHAR CARD

C : The tourist has a criminal record

G : The tourist is the government employee

V : The tourist has repeated the resort visit in a span of six months

Output : F : The tourist would be allowed to book the resort at rebate.

[1 indicates Yes and 0 indicates No].

Draw the truth table for inputs and outputs given above and write the ***Cardinal Sum of product expression*** for F (A, C, G, V). [5]

(ii) (a) Reduce the above expression F (A, C, G, V) by using 4-variable Karnaugh map, showing the various groups (i.e. octal, quads and pairs). [4]

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

Question 4

(i) Given the Boolean function $F(P, Q, R, S) = \pi(2, 3, 5, 7, 8, 10, 11, 12, 13, 15)$.

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

(b) Draw the logic gate diagram for the reduced expression using NOR gate only. Assume that the variables and their complements are available as inputs. [1]

(ii) Simplify the following expression using Boolean laws : [2]

$F = PQ + (P + Q) \cdot (P + PR) + Q$

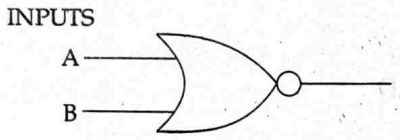
(iii) Verify if the following proposition is valid using the truth table: [3]

$A \Rightarrow (B \wedge C) = (A \Rightarrow B) \wedge (B \Rightarrow C)$

Question 5

(i) What is a decoder? How is it different from a multiplexer? Draw the logic circuit for a 2 to 4 decoder and explain its working. [5]

(ii) Answer the following questions related to the below image :



(a) What is the output of the above gate if input A = 0, B = 1? [1]

(b) What are the values of the inputs if output = 1? [1]

(iii) Draw a truth table with a 3 input combination which outputs 1 if there are odd number of 0's. Also derive a SOP expression for the output. [3]

SECTION - B

Answer any two questions

Each program should be written in such a way that its clearly depicts the logic of the problem.

This can be achieved by using mnemonic name and comments in the program.

(Flowcharts and Algorithms are **not** required)

The programs must be written in Java.

Question 6 ***[10]***

Design a class MatRev to reverse each element of a matrix.

Example :

72	371	5
12	6	426
5	23	94

becomes

27	173	5
21	6	624
5	32	49

Some of the members of the class are given below :

Class name : ***MatRev***

Data members/instance variables:

arr[ ][ ]	:	to store integer elements
m	:	to store the number of rows
n	:	to store the number of columns

Member functions/methods :

MatRev(int mm, int nn)	:	parameterized constructor to initialise the data members m = mm and n = nn
void fillarray( )	:	to enter elements in the array
int reverse(int x)	:	returns the reverse of the number x
void revMat(MatRev P)	:	reverses each element of the array of the parameterized object and stores it in the array of the current object
void show( )	:	displays the array elements in matrix form

Define the class MatRev giving details of the constructor ( ), void fillarray (), int reverse(int), void revMat(MatRev) and void show(). Define the main () function to create objects and call the functions accordingly to enable the task.

Question 7 ***[10]***

Given are two strings, input string and a mask string that remove all the characters of the mask string from the original string.

Example : INPUT: ORIGINALSTRING : ***communication***

MASK STRING : ***mont***

OUTPUT : ***cuicai***

A class ***StringOp*** is defined as follows to perform above operation. Some of the members of the class are given below :

Class name : ***StringOp***

Data members/instance variables :

str	:	to store the original string
msk	:	to store the mask string
nstr	:	to store the resultant string

Methods / Member functions:

StringOp( )	:	default constructor to initialize the data member with legal initial value
void accept( )	:	to accept the original string str and the mask string msk in lower case
void form( )	:	to form the new string nstr after removal of characters present in mask from the original string
void display( )	:	to display the original string and the newly formed string nstr

Specify the class StringOp giving details of the constructor( ), void accept( ), void form( ) and void display( ). Define a main( ) function to create an object and call all the functions accordingly to enable the task.

Question 8 ***[10]***

Design a class Pronic to check if a given number is a pronic number or not.
[A number is said to be pronic if the product of two consecutive numbers is equal to the number]

Example : $0 = 0 \times 1$

$2 = 1 \times 2$

$6 = 2 \times 3$

$12 = 3 \times 4$

thus, 0, 2, 6, 12... are pronic numbers.

Some of the members of the class are given below :

Class name	:	***Pronic***
Data members/instance variables :		
nnum	:	to store a positive integer number
Methods / Member functions :		
Pronic( )	:	default constructor to initialize the data member with legal initial value
void acceptnum( )	:	to accept a positive integer number
boolean ispronic(int v)	:	returns true if the number 'num' is a pronic number, otherwise returns false using recursive technique.
void check( )	:	checks whether the given number is a pronic number by invoking the function ispronic( ) and displays the result with an appropriate message.

Specify the class Pronic giving details of the constructor( ), void acceptnum( ), boolean ispronic(int) and void check( ). Define a main( ) function to create an object and call the functions accordingly to enable the task.

SECTION - C

Answer only **two** questions.

Each program should be written in such a way that its clearly depicts the logic of the problem stepwise.

This can also be achieved by using comments in the program and mnemonic names or pseudocodes for algorithms. The program must be written in Java and the algorithms must be written in general/standard form, wherever required/ specified.

(Flowcharts are **not** required)

Question 9

Recycle is an entity which can hold at the most 100 integers. The chain enables the user to add and remove integers from both the ends i.e. front and rear.

Define a class ***ReCycle*** with the following details:

Class name	:	***ReCycle***
Data members/instance variables:		
ele[ ]	:	the array to hold the integer elements
cap	:	stores the maximum capacity of the array
front	:	to point the index of the front
rear	:	to point the index of the rear
Methods / Member functions :		
ReCycle (int max)	:	constructor to initialize the data cap = m rear = 0 and to create the integer array.
void pushfront(int v)	:	to add integers from the front index if possible else display the message("full from front").
int popfront( )	:	to remove the return elements from front. If array is empty then return –999.
void pushrear(int v)	:	to add integers from the front index if possible else display the message("full from rear").
int poprear( )	:	to remove and return elements from rear. If the array is empty then return-999.

(i) Specify the class ***ReCycle*** giving details of the functions ***void pushfront(int) int poprear( )***. Assume that the other functions have been defined.The main( ) funct algorithm need NOT be written [4]

(ii) Name the entity described above and state its principle. [1]

Question 10 [5]

A library issues books on rental basis at a 2% charge on the cost price of the book per day. As per the rules of the library, a book can be retained for 7 days without any fine. If the book is returned after 7 days, a fine will also be charged for the excess days as per the chart given below :

Number of excess days	Fine per day (₹)
1 to 5	2.00
6 to 10	3.00
Above 10	5.00

A super class ***Library*** has been defined. Define a sub class ***Compute*** to calculate the fine and the total amount. The details of the members of both the classes are given below.

Class name : Library

Data members/instance variables:

name	:	to store the name of the book
author	:	to store the author of the book
p	:	to store the price of the book (in decimals)

Methods / Member functions:

Library(...)	:	parameterized constructor to assign values to the data members
void show( )	:	displays the book details

Class name : Compute

Data members/instance variables:

d	:	number of days taken in returning the book
f	:	to store the fine (in decimals)

Methods / Member functions:

Compute(...)	:	parameterized constructor to assign values to the data members of both the classes
void fine( )	:	calculates the fine for the excess days as given in the table above
void show( )	:	displays the book details along with the number of days, fine and the total amount to be paid. Total amount is (2% of price of book * total no of days) + fine

Assume that the super class ***Library*** has been defined. Using the ***concepts of Inheritance***, specify the class ***Compute*** giving the details of ***constructor***, ***void fine ( )*** and ***void show ( )*** functions.

The super class, main function and algorithm need NOT be written.

Question 11

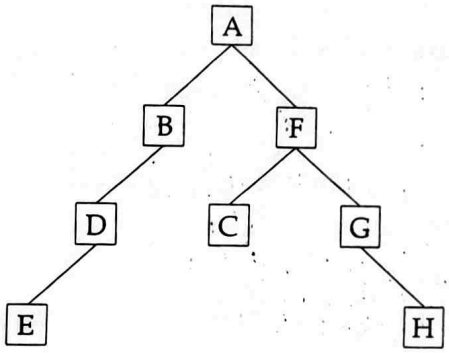
(i) A linked list is formed from the objects of the class :

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

Write an Algorithm OR a Method to insert a node at the beginning of an existing linked list. The method declaration is as follows :

void InsertNode(Nodes starPtr, int n) [2]

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



(a) Write the post order traversal of the above tree structure. [1]

(b) Name the children of the nodes B and G. [1]

(c) State the root of the right sub tree. [1]

#####

Q1

i) b) X.Y.Z. [1 Mark]

ii) b) Both Assertion and Reason are true but Reason is not the correct explanation for Assertion. [1Mark]

iii) c) 0 [1 Mark]

iv) c) Commutative Law [1 Mark]

v) a) Both Assertion and Reason are true but Reason is not the correct explanation for Assertion [1 Mark]

vi) c) A'+B+C+D' [1 Mark]

vii) AND [1 Mark]

viii) O (n) It is represents that the algorithm takes at the most n steps while(n^2) represents that the algorithm takes n^2 steps at the most steps. [1 Mark]

ix) 2 AND gates, 1 OR gate and 2 XOR gates [1 Mark]

x) Stack is a LIFO (Last In First Out) and Queue is FIFO(First in First Out) data structure.

Q2 :

i) **Interface :** An interface defines a protocol of common behavior. The aim of interface is to dictate common behavior among objects from diverse classes. Interface can have only two things constant variables and abstract methods while class contain data member and member function with code.

ii) 3952 [1Mark for Answer + 1 Mark for working]

iii) a) Ans : 3 [1Mark for Answer + 1 Mark for working]

b) Binary Search [1 Mark]

iv) ?1? = 0 , ?2? = 10 ,?3? = n%2 [1 Mark each]

Q3

A	C	G	V	F
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	1
1	0	0	0	1
1	0	0	1	1
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	1
1	1	1	0	0
1	1	1	1	1

Marking:

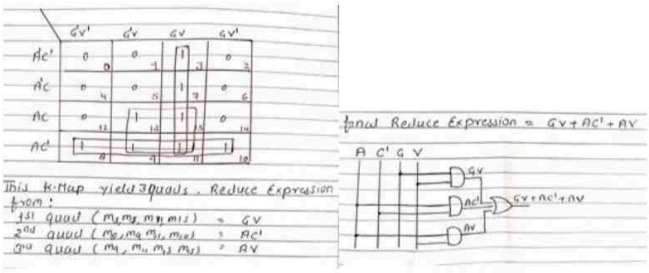
16 Outputs of Truth Table (¼ for each output = 4 Marks)

+ 1 Mark for SOP Expression .

1 for correct form of K – Map + 3 Marks for reduced Expression + 1 Mark for Logic Gate

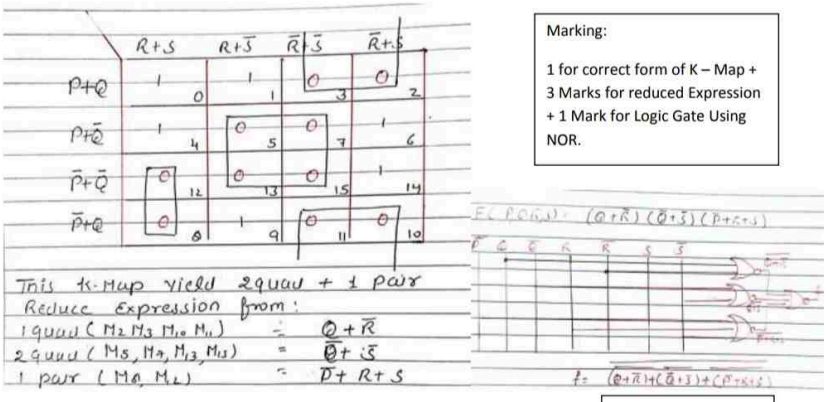
X (A,C,G,V) = ∑ (3,7,8,9,10,11,13,15)

ii)



Question 4

i)



Marking:

1 for correct form of K – Map + 3 Marks for reduced Expression + 1 Mark for Logic Gate Using NOR.

(ii)

F = PQ+(P+Q).(P+PR) +Q

= PQ+PP + PP R +PQ +PQR +Q [DISTRIBUTIVE LAW]

= PQ+P+PR+PQ+PQR+Q [IDEMPOTANT LAW]

= P(Q+1+R+Q+QR) +Q [DISTRIBUTIVE LAW]

= P.1+Q [PROPERTY OF 1]

= P+Q [PROPERTY OF 1]

Marking:

1 for Answer

½ – Laws

½ - Working

(iii)

A	B	C	B^C	A=>B^C	A=>B	B=>C	(A=>B)^(B=>C)
0	0	0	0	1	1	1	1
0	0	1	0	1	1	1	1
0	1	0	0	1	1	0	0
0	1	1	1	1	1	1	1
1	0	0	0	0	0	1	0
1	0	1	0	0	0	1	0
1	1	0	0	0	1	0	0
1	1	1	1	1	1	1	1

½ : B^C

½ : A=B^C

½ : A=>B

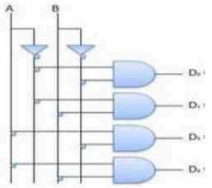
½ : B=>C

½ : (A=>B)^(B=>C)

½ : Not Valid

Question 5.

(i) A Decoder is combinational circuit converts Binary number in to Decimal number.



Marking: (Total 5 Marks)

1 DECODER

1– DIFFERENCE

2 - LOGIC GATE

1-- WORKING

Multiplexer uses 2 to 4 Decoder to control 4 input signals. Multiplexer always provide one output while decoder provide 2^n output according n inputs.

Working : 2 to 4 decoder is used to convert two binary number(00,01,10,11) in to decimal number (0,1,2,3)

(ii) (a) 0 [1 Mark] (b) A=0 & B=0 [1 Mark]

A	B	C	F(A,B,C)
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	0

Marking: (For each 1)

1/2 *4 = 2 Marks

1– SOP

SOP Expression : A'B'C' +A'BC+AB'C+ABC' OR $\sum$ (1,3,5,7)

Marking Scheme of Section B

Question 6

```
import java.util.*;
class MatRev
{
int arr[][],m,n;
MatRev(int mm,int nn)
{
m=mm;
n=nn;
arr=new int[m][n];
  }
  void fillarray()
  {
     Scanner sc=new Scanner(System.in);
 System.out.println("enter "+m+"*"+n+" matrix value");
     for(int i=0;i<m;i++)
     {
        for(int j=0;j<n;j++)
        arr[i][j]=sc.nextInt();
     }
  }
  int reverse(int x)
  {
    int rev=0;
     for(int i=x;i!=0;i/=10)
     {
        rev = rev*10+i%10;
     }
```

Question 6 MatRev (Marking)

Class + DM - 1

MatRev(int mm,int nn) - 1

void fillarray() - 1

int reverse(int x) - 1

void revMat(MatRev P) - 3

void show() - 1

void main() - 1

VD Table - 1

```
    return rev;
  }
  void revMat(MatRev P)
  {
      for(int i=0;i<P.m;i++)
      {
       for(int j=0;j<P.n;j++)
        {
       this.arr[i][j]=reverse(P.arr[i][j]);
        }
      }
  }
      void show()
      {
        for(int i=0;i<m;i++)
        {
          for(int j=0;j<n;j++)
          {
          System.out.print(arr[i][j]+"\t");
          }
          System.out.println();
        }
       }
     public static void main(int m,int n)
      {
        MatRev ob1=new MatRev(m,n);
        MatRev ob2=new MatRev(m,n);
        ob1.fillarray();
        ob2.revMat(ob1);
```

```
        System.out.println("orignal matrix");
        ob1.show();
       System.out.println("reverse matrix");
        ob2.show();
    }
   }
```

Question 7

```
import java.util.*; class
StringOp
{
   String str;
   String nstr;
   String msk;
   Scanner sc=new Scanner(System.in);
   StringOp()
   {
      str="";
      nstr="";
      msk="";
   }
   void accept()
   {
      System.out.println("Enter the original word");
      str=sc.next()+sc.nextLine();
       System.out.println("enter the mask string");
      msk=sc.next();
   }
   void form()
   {
      int l1=str.length();int l2=msk.length();
      for (int i=0;i<l1;i++)
      {
         char c1=str.charAt(i);
         if (msk.indexOf(c1)==-1)
             nstr=nstr+c1;
         }
      }
   void display()
   {
       System.out.println("original string: "+str);
       System.out.println("changed string: "+nstr);
    }
   public static void main()
   {
```

Question 6 StringOP(Marking)	
Class + DM - 1	
StrinOp () -	1
void accept() - 1	
void form() -	4
void display() - 1	
void main() -	1
VD Table -	1

```
        StringOp ob=new StringOp();
        ob.accept();
        ob.form();
        ob.display();
    }
Question 8
import java.util.*;
class Pronic
{
int num;
Pronic()
{
num=0;
}
void acceptnum()
{
Scanner sc=new Scanner(System.in);
System.out.println("Enter a number");
num=sc.nextInt();
}
boolean ispronic(int v)
{
int i=0;
while(i*(i+1) < v)
{
i++;
 }
if(i*(i+1)==v)
return true;
else
return false;
}
void check()
{
if(ispronic(num))
System.out.println(num+" is pronic number");
else
System.out.println(num+" is not a pronic number");
}
public static void main()
{
Pronic ob= new Pronic();
ob.acceptnum();
ob.check();
}
}//class
```

Question 7 Pronic (Marking)

Class + DM - 1

Pronic () - 1

void acceptnum() – 1

boolean ispronic(int v) - 4

void check() - 1

void main() - 1

VD Table - 1

Marking Scheme of Section C

```
Question 9
class ReCycle
{
int ele[],cap,front,rear;
void pushFront(int v)
 {
    if(front==0 && rear !=front)
    System.out.println("full from front");
    else
    {
       if(front==rear)rear=1;
       else front--;
       ele[front] = v;
    }
   }
   int poprear()
   {
    if( front == rear)
    return -999;
    else
    {
       int val=ele[--rear];
       ele[rear]=0;
       if(front==rear)front=rear=0;
       return val;
     }
 }
 }
```

Q9 MARKING

Class + Data Member - 1 Mark

void pushFront(int v) – 1 ½ Marks

int poprear() – 1 ½ Marks

Entity Name – 1 Mark

ii) Entity Name is Double Ended Queue (Deque or Dque)

Question 10

```
class Compute extends Library
{
int d,f;
Compute(String n,String a,double p,int d1)
{
super(n,a,p);
d=d1;
f=0;
   }
void fine()
{
int x=d-7;
if(x>=1 && x<=5 ) f=x*2;
else if( x>=6 && x<=10)
```

Question 10 Marking

1. extends keyword - ½ marks
2. class + Data member - ½ marks
3. Compute(String n,String a,double p,int d1)
   super(...) ½ Mark
   Initialize D & F ½ Mark
4. void fine () - [2 marks].
5. void show() 1 Mark

```
f= 5*2+(x-5)*3;
else if(x>10)
f= 5*2+ 5*3+(x-10)*5;
}
void show()
{
fine();
double amt= super.p *.02 *d + f;
super.show();
System.out.println(" days = "+d);
System.out.println(" fine = "+f);
System.out.println(" amount = "+amt);
   }
}
```

Question 11

i)

```
void InsertNode(Node starPtr , int n)
{
        Node nptr = new Node();
        nptr.num=n;
        nptr.next= starPtr;
        starPtr= nptr;
}
```

Or Algorithm with above mention 4 steps

Marking (for each steps)
½ * 4 = 2 Marks

ii)

a) EDBCHGFA [1 Mark]

b) Children of B is D , & children of G is H [1 Mark]

c) F [1 Mark]