Q1. Law of excluded middle:
A v ~A = TRUE

Q2. Give the use and some functions of the File class? Ans: File class present in io package is used to store and wok upon a file name.
File file=new File("A.DAT");
Functions: file.exists(), file.length()
file.renameTo(new File("c:\\B.DAT"));//can also move a file
file.delete();, file.isDirectory();
String[] file.list();

Q3. Give an example of reading a text file using a scanner class.
Scanner sc=new Scanner(new FileReader("a.txt"));
while (scanner.hasNext()) System.out.println(scanner.next());

Q4. Give the steps for printing in Java.

```
import java.awt.*;
import java.awt.print.*;
public class Printing implements Printable
{
public static void main(String args[])

{   //[ Step 1] Create a print Job
    PrinterJob printJob=PrinterJob.getPrinterJob();

    //[ Step 2] Render content to page
    printJob.setPrintable(new Printing());

    //[ Step 3] Display print dialog
    if(printJob.printDialog())
    {   try{    printJob.print();
          }
        catch(Exception PrintException)  {}
    }
}//main

//print() is a method in printable
public int print(Graphics g, PageFormat pageFormat, int page)
{   Graphics2D g2d= (Graphics2D)g;
    g2d.setColor(Color.black);
    if(page==0)//First Page
    {   g2d.drawString("Java",50,50);
        return(PAGE_EXISTS);
    }
    else
        return(NO_SUCH_PAGE);
}//print
}//class:
```

Q5. Give the Towers of Hanoi Program

```
class TowersOfHanoi
{ public static void main (String args[])
  { move(4, 3, 1, 2); //disks, to, from, temp
  }
  static void move(int n, int to, int from, int temp)
  { if (n > 0)
    {   move(n-1, temp, from, to);
        System.out.println("move "+from+" -> "+to);
        move(n-1, to, temp, from);
    }//if
  }//move
}//class
```

Q6. Give a program to solve a Maze.

```
//A Basic Maze (only for orientation, needs modifications)
//Objective: move from start(0,0) to end(4,4)
//'1'=open, '0'=closed, '.'=travelled
public class Maze
{
   char[][] maze= {
           {'1','0','1','0','1'},
           {'1','1','1','1','0'},
           {'0','0','0','1','1'},
           {'1','0','1','1','0'},
           {'1','0','1','1','1'}
    };
    boolean solved=false;

    void display()
    {   for(int i=0; i<maze.length; i++)
        {   for(int j=0; j<maze[i].length; j++)
               System.out.print(maze[i][j]+" ");
            System.out.println();
        } System.out.println();
    }//display()

    void move(int r, int c)
    {   if(r<0 || r>maze.length-1 || c<0 || c>maze.length-1 || maze[r][c]!='1') return;
        maze[r][c]='.';
        if(r==maze.length-1 && c==maze.length-1)
        {   solved=true;
            return;
        }
        if(!solved) move(r+1,c);
        if(!solved) move(r,c+1);
        if(!solved) move(r-1,c);
        if(!solved) move(r,c-1);
    }//move()
```

```
  public static void main(String args[])
  {   Maze obj=new Maze();
      obj.display();
      obj.move(0,0);
      obj.display();
   }//main
}//class
```

Q7. Algorithm to merge sort a linked list

MergeSort(Start)
1) If head is NULL or there is only one element in the Linked List
   then return.
2) Else divide the linked list into two halves.
     Split(Start); /* a and b are two halves */
3) Merge Sort (recursion) the two halves a and b.
     MergeSort(a);
     MergeSort(b);
4) Merge the sorted a and b and update Start.
    Start = SortedMerge(a, b);

Q8. Describe the use of the StreamTokenizer class in reading a text file.

1. StreamTokenizer tokenizes Reader classes.
2. This process is called “Lexing”.
3. nextToken() brings the next token in the object.
4. ttype helps us find know the type of token.
5. sval() Returns string value of the token.
6. nval()Returns numeric value of the token.

```
//Processing a file with Stream Tokenizer
   public static void read()throws IOException
   {   FileReader fr=new FileReader("S.DAT");
       StreamTokenizer st=new StreamTokenizer(fr);
       int c1=0, c2=0, c3=0, c4=0;
       int tokenType=0;
       do
       {   tokenType=st.nextToken();
          switch(tokenType)
          {   case StreamTokenizer.TT_NUMBER:
                    System.out.println(st.nval);
                    break;
              case StreamTokenizer.TT_WORD:
                    System.out.println(st.sval);
                    break;
          }//switch
       }while(tokenType!=StreamTokenizer.TT_EOF);
       fr.close();
   }//read
```

END