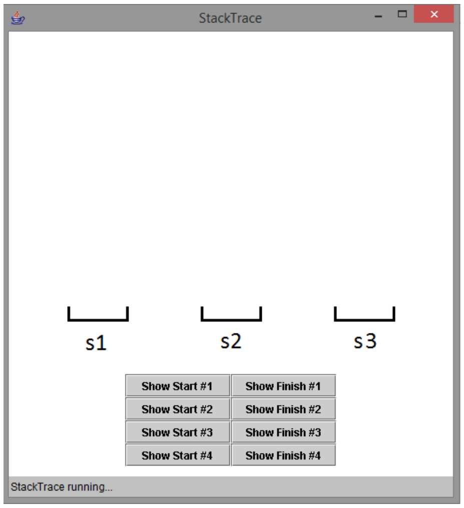
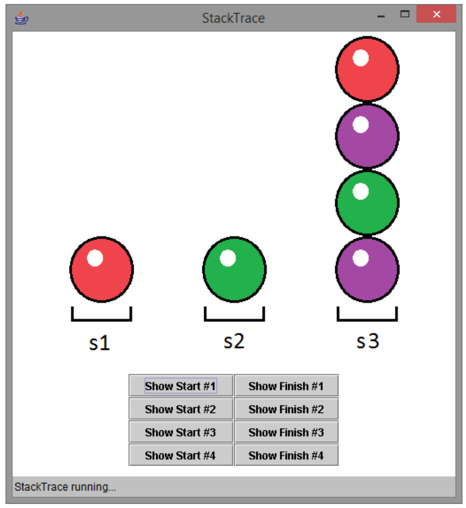
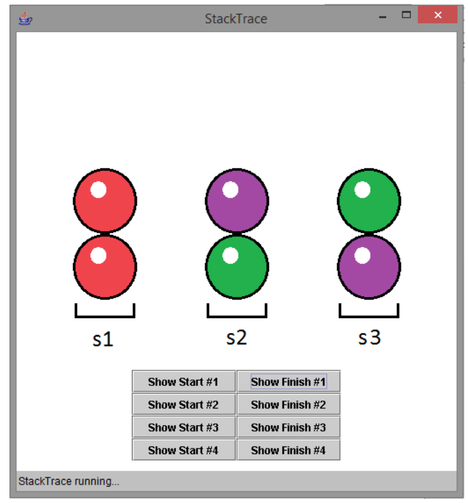
Stack Trace



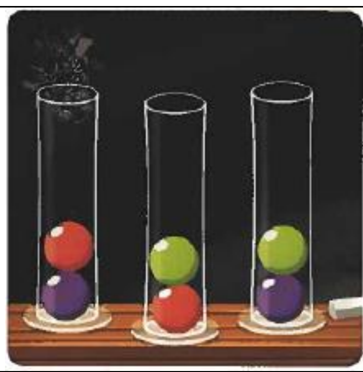
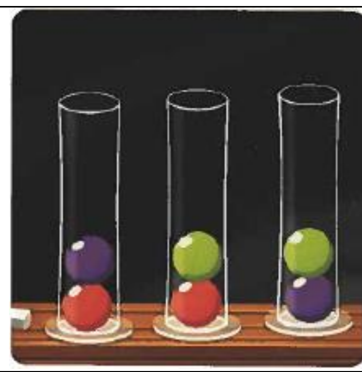
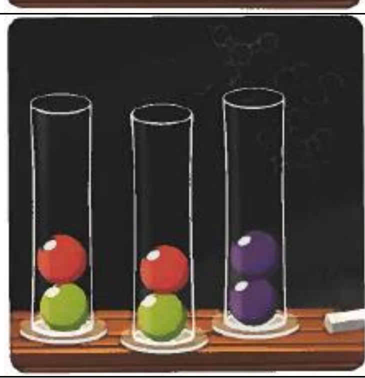
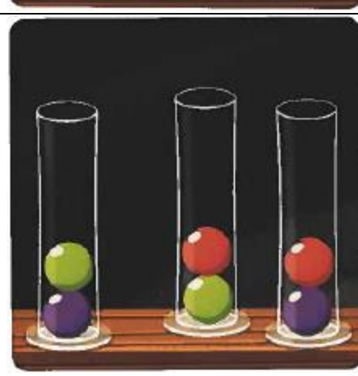


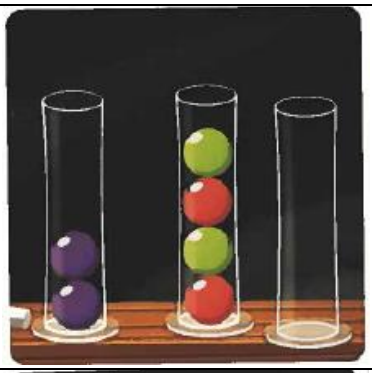
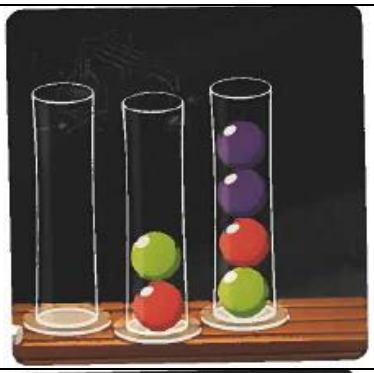
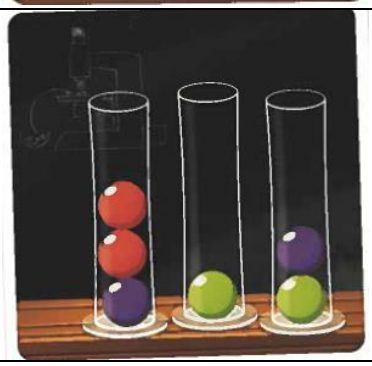
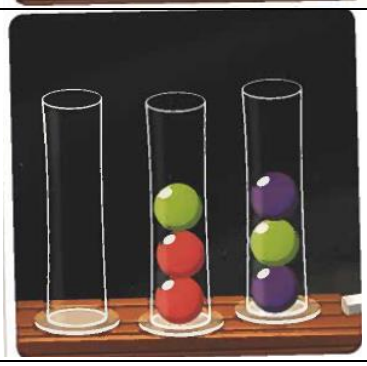


You will need to create a file for the program (starter code below), you will need 5 pictures too.

Your job is to load the 3 Stacks to create the following initial puzzles.

- You will be coding the Stacks in actionPerformed.
- When the “Show Start #1” button is pressed, it should show the stacks in the position of puzzle #1 below. Make sure that you clear the Stack before you load the puzzles.
- When the “Show Finish #1” button is pressed, it should show the stacks in the answer of puzzle #1 below. You should push and pop the elements from the initial puzzles to get to the answer.
- Complete all four puzzles.

	Start	Finish
#1		
#2		

	Start	Finish
#3		
#4		

Starter Code:

```
import javax.swing.*;
import java.awt.*;
import java.awt.event.*;
import java.applet.Applet;
import java.util.*;

public class StackTrace extends Applet implements ActionListener
{
    JLabel stackPic[] = new JLabel [35];
    Stack s1 = new Stack ();
    Stack s2 = new Stack ();
    Stack s3 = new Stack ();

    public void init ()
    {

        resize (500, 500);
        Panel p = new Panel (new GridLayout (5, 7));
        for (int i = 0 ; i < 35 ; i++)
        {
            stackPic [i] = new JLabel (createImageIcon ("b.png"));
            stackPic [i].setPreferredSize (new Dimension (75, 75));
            p.add (stackPic [i]);
        }
        add (p);
        stackPic [29].setIcon (createImageIcon ("s1.png"));
        stackPic [31].setIcon (createImageIcon ("s2.png"));
        stackPic [33].setIcon (createImageIcon ("s3.png"));

        JButton b[] = new JButton [8];
        Panel p2 = new Panel (new GridLayout (4, 2));
        int num = 1;
        for (int i = 0 ; i < 8 ; i += 2)
```

```
    {
        b [i] = new JButton ("Show Start #" + num);
        b [i].setActionCommand ("Start" + num);
        b [i].addActionListener (this);
        p2.add (b [i]);

        b [i + 1] = new JButton ("Show Finish #" + num);
        b [i + 1].setActionCommand ("Finish" + num);
        b [i + 1].addActionListener (this);
        p2.add (b [i + 1]);
        num++;
    }
    add (p2);
}


public void actionPerformed (ActionEvent e)
{
    if (e.getActionCommand ().equals ("Start1"))
    {
        clearAllStacks ();
        s1.push ("r");
        s2.push ("g");
        s3.push ("p");
        s3.push ("g");
        s3.push ("p");
        s3.push ("r");
    }
    else if (e.getActionCommand ().equals ("Finish1"))
    {
        s1.push (s3.pop ());
        s2.push (s3.pop ());
    }
    else if (e.getActionCommand ().equals ("Start2"))
    {
    }
    else if (e.getActionCommand ().equals ("Finish2"))
    {
    }
    else if (e.getActionCommand ().equals ("Start3"))
    {
    }
    else if (e.getActionCommand ().equals ("Finish3"))
    {
    }
    else if (e.getActionCommand ().equals ("Start4"))
    {
    }
    else if (e.getActionCommand ().equals ("Finish4"))
    {
    }
    viewAllStacks ();
}


public void clearAllStacks ()
{
    s1.clear ();
    s2.clear ();
    s3.clear ();
}


public void viewAllStacks ()
{
    for (int i = 0 ; i < 27 ; i++)
        stackPic [i].setIcon (createImageIcon ("b.png"));
```

```
        drawStack (s1, 22);
        drawStack (s2, 24);
        drawStack (s3, 26);
    }


    public void drawStack (Stack s, int start)
    {
        Stack s4 = new Stack ();

        while (!s.isEmpty ())
            s4.push (s.pop ());

        while (!s4.isEmpty ())
        {
            String value = (String) s4.pop ();
            stackPic [start].setIcon (createImageIcon (value + ".png"));
            s.push (value);
            start -= 7;
        }

    }


    protected static ImageIcon createImageIcon (String path)
    {
        java.net.URL imgURL = StackTrace.class.getResource (path);
        if (imgURL != null)
        {
            return new ImageIcon (imgURL);
        }
        else
        {
            System.err.println ("Couldn't find file: " + path);
            return null;
        }
    }
} //end applet
```