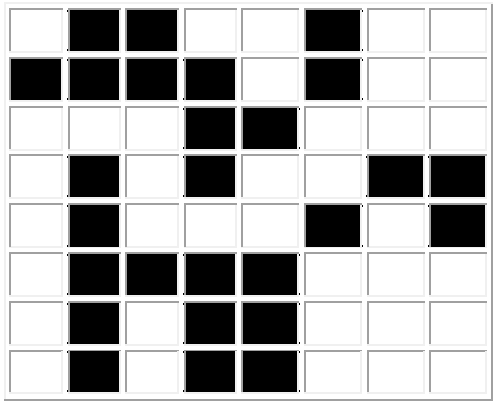
Assignment 4

Due October 3

Program 1.

An n by n grid consists of black and white squares as shown below. Your problem is to determine the number of connected white areas in the grid and the size (number of squares) in each area.



For example, grid shown above has 5 white areas of sizes 1,3,21,10 and 2 cells.

You should represent the grid with an array. Each entry in the array should be

either b or w. Like the maze problem, include a border of black squares around the array.

For example the above square can be represented as

```
b b b b b b b b b b
b w b b w w b w w b
b b b b b w b w w b
b w w w b b w w w b
b w b w b w w b b b
b w b w w w b w b b
b w b b b b w w w b
```

```
b w b w b b w w w b
b w b w b b w w w b
b b b b b b b b b b
```

The first line of input to your program should be n, the size of the array. This should be followed by the array of b's and w's. Input should be from a file called grid.txt.

Remember if the grid is n by n, the array (with its extra border) is (n+2) by (n+2).

The solution to this requires a recursive divide and conquer algorithm. Go through the board cell by cell. Use four recursive calls and mark a cell as visited when it has been counted.

Output is a list of white areas and the size of each.

This is a very, very short recursive program and can be done in about 5 lines. Think about the divide and conquer car parking program

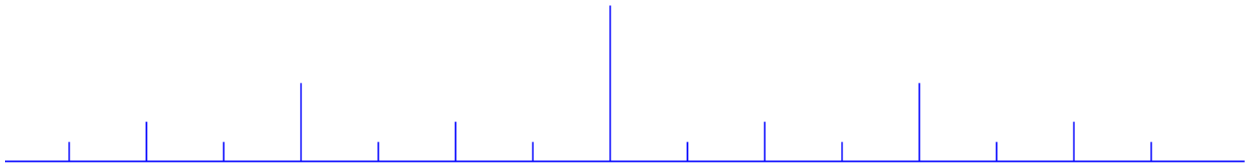
Program 2:

This is problem 9 on page 231 in the textbook. Here is apicture of the problem from the text:

If you do not have the text here is the problem:

Write a recursive method that draws a picture of a 12-inch ruler. Mark inches, half inches, quarter inches and eighth inches. Mark the half inches with marks that are smaller than those that mark the inches. Mark the qy=uarter inches with marks that are smaller than those that mark the half inches and so on. Your picture need not be full size. Hint: Draw a mark in the middle of the ruler and then draw rulers to the left and right of the mark

This is another divide and conquer problem. It is very short. The output should look like this:



Here is a sketch of the program...You just need to fill in the backtracking method

```
import java.awt.*;
import javax.swing.*;
public class Ruler extends JPanel
{
    public void paintComponent(Graphics g)
    {
      super.paintComponent(g);
      setBackground(Color.WHITE);
      g.setColor(Color.BLUE);
      g.drawLine(0,200, 810,200);     // the horizontal line in the picture
      addMark(0,800,100, g);           //between 0 and 800  of  height 100
   }
   public void addMark(int lo,int hi, int size, Graphics g)
   {
            //your code goes here

   }

   public static void main(String[] args)
  {
      JFrame f = new JFrame();
      f.setBounds(0,0, 850, 500);
      Ruler r = new Ruler();
      f.add(r);
      f.setVisible(true);
   }
}
```