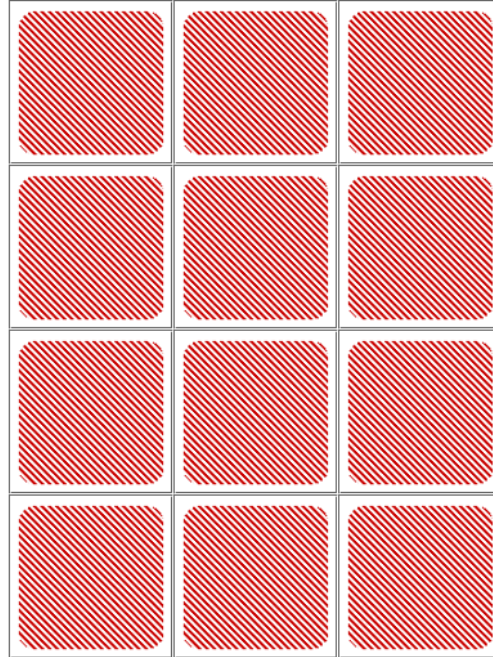


# Halloween Matching

1. Add the array to make these pictures.
2. Add a reset button. Change its font and color. Its actionPerformed must be "Reset".
3. Add a title. Change its font.



## Starter code:

```
import javax.swing.*; import java.applet.*; import java.awt.event.*; import java.awt.*;
public class halloween extends Applet implements ActionListener
{
    int hallow[] [] = ?
    int rows = 4;
    int cols = 3;
    JButton pics[] = new JButton [rows * cols];

    public void init ()
    {
        Panel grid = new Panel (new GridLayout (rows, cols));
        int m = 0;
        for (int i = 0 ; i < rows ; i++)
        {
            for (int j = 0 ; j < cols ; j++)
            {
                pics [m] = new JButton (createImageIcon ("back.png"));
                pics [m].addActionListener (this);
                pics [m].setActionCommand (m + "");
                pics [m].setPreferredSize (new Dimension (128, 128));
                grid.add (pics [m]);
            }
        }
    }
}
```

```

        m++;
    }
}
add (grid);
}

public void actionPerformed (ActionEvent e)
{
    if (e.getActionCommand ().equals ("Reset"))
    {
        for (int m = 0 ; m < pics.length ; m++)
            pics [m].setIcon (createImageIcon ("back.png"));
    }
    else
    {
        int n = Integer.parseInt (e.getActionCommand ());
        int x = n % rows;
        int y = n / rows;
        pics [n].setIcon (createImageIcon ("hal" + hallow [x] [y] + ".png"));
    }
}

protected static ImageIcon createImageIcon (String path)
{
    java.net.URL imgURL = halloween.class.getResource (path);
    if (imgURL != null)
        return new ImageIcon (imgURL);
    else
        return null;
}
}

```