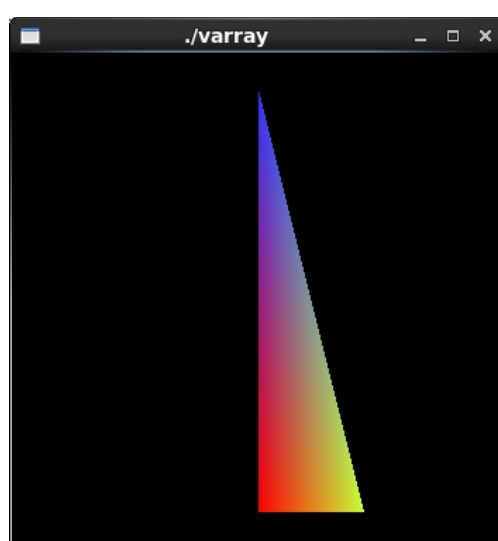
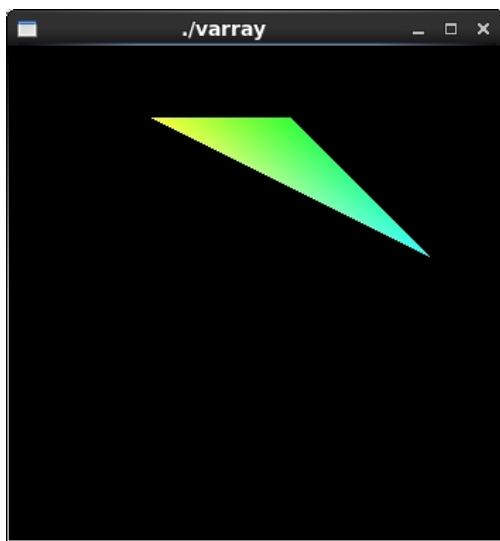
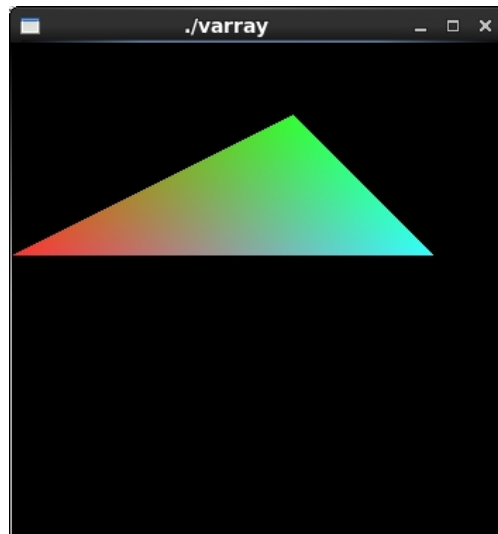
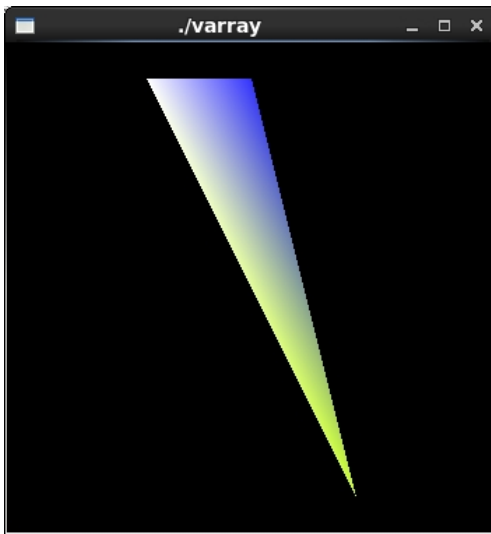


Modify the program so that it displays other triangular shapes when you click the mouse buttons.



```
...  
void display(void)  
{  
    glClear (GL_COLOR_BUFFER_BIT);  
    if (derefMethod == DRAWARRAY)  
        glDrawArrays (GL_TRIANGLES, 0, 6);  
    else if (derefMethod == ARRAYELEMENT) {  
        glBegin (GL_TRIANGLES);  
        glVertex (2); //note: vertices 1, 3, 5 is a straight line  
        glVertex (3);  
        glVertex (4);  
        glEnd ();  
    }  
    else if (derefMethod == DRAWELEMENTS) {
```

```
//  GLuint indices[4] = {0, 1, 3, 4};
    GLuint indices[3] = {1, 3, 4};

//  glDrawElements (GL_POLYGON, 4, GL_UNSIGNED_INT, indices);
    glDrawElements (GL_TRIANGLES, 3, GL_UNSIGNED_INT, indices);
}
glFlush ();
}
...
```

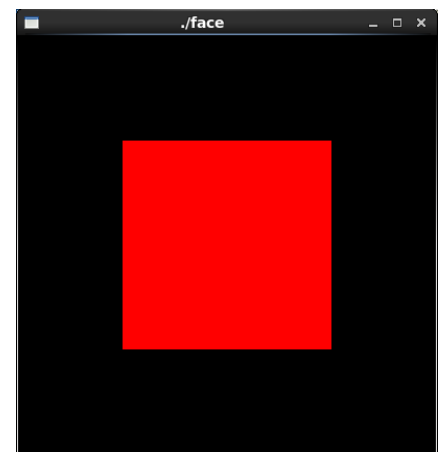
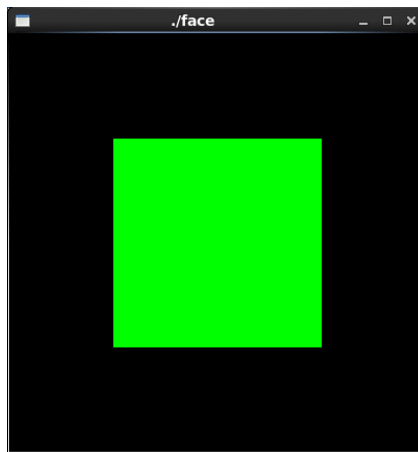
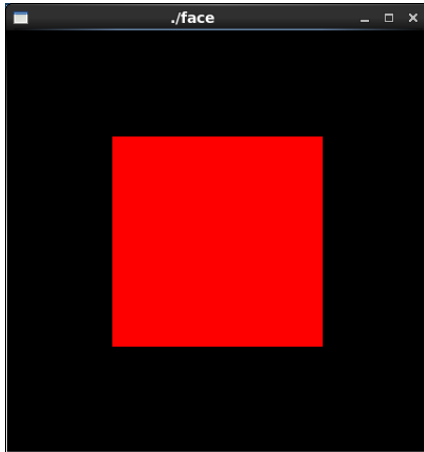
Then rewrite the routines using the `glArrayElement()` function (see class notes).

```
...
void display(void)
{
    glClear (GL_COLOR_BUFFER_BIT);
    if (derefMethod == DRAWARRAY) {
        //glDrawArrays (GL_TRIANGLES, 0, 6);
        glBegin( GL_TRIANGLES );
        for ( int i = 0; i < 6; ++i )
            glArrayElement ( i );
        glEnd();
    }
    else if (derefMethod == ARRAYELEMENT) {
        glBegin (GL_TRIANGLES);
        glArrayElement (2); //note: vertices 1, 3, 5 is a straight line
        glArrayElement (3);
        glArrayElement (5);
        glEnd ();
    }
    else if (derefMethod == DRAWELEMENTS) {
        glBegin ( GL_POLYGON );
        glArrayElement (0);
        glArrayElement (1);
        glArrayElement (3);
        glArrayElement (4);
        glEnd();
    }
    /*  GLuint indices[4] = {0, 1, 3, 4};
        GLuint indices[3] = {1, 3, 4};

        glDrawElements (GL_POLYGON, 4, GL_UNSIGNED_INT, indices);
        glDrawElements (GL_TRIANGLES, 3, GL_UNSIGNED_INT, indices);
    */
}
glFlush ();
}
...
```

(Extra Credit 5 points) Make use of the functions `glPolygonMode()` and `glCullFace()` discussed in Chapter 3 (Drawing Objects) to write a program that can display the front face and back face of a square alternately:

- 1. It displays the front face in red.*
- 2. When clicking the mouse at the square, it displays the back face in green.*
- 3. When clicked again at the square, it displays the front face again, and so on.*



```
#include <GL/glut.h>
#include <stdlib.h>
#include <stdio.h>

#define POINTER 1
#define INTERLEAVED 2

#define DRAWARRAY 1
#define ARRAELEMENT 2
#define DRAWELEMENTS 3

//int setupMethod = POINTER;
int derefMethod = DRAWARRAY;

void display(void)
{
    glClear (GL_COLOR_BUFFER_BIT);
    glEnable (GL_CULL_FACE);
    if (derefMethod == DRAWARRAY) {
        glCullFace ( GL_BACK );
        glColor3f( 1.0, 0.0, 0.0 );
        glFrontFace( GL_CCW );
        glPolygonMode( GL_FRONT, GL_FILL );
        glBegin( GL_POLYGON );
        glVertex2i( 100, 100 );
        glVertex2i( 300, 100 );
        glVertex2i( 300, 300 );
        glVertex2i( 100, 300 );
```

```

    glEnd();
}
else if (derefMethod == ARRAYPELEMENT) {
    glCullFace (GL_FRONT);
    glColor3f( 0.0, 1.0, 0.0 );
    glFrontFace( GL_CCW );
    glPolygonMode( GL_BACK, GL_FILL );
    glBegin( GL_POLYGON );
        glVertex2i( 100, 100 );
        glVertex2i( 100, 300 );
        glVertex2i( 300, 300 );
        glVertex2i( 300, 100 );
    glEnd();
}
glFlush ();
}

void init(void)
{
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glShadeModel (GL_SMOOTH);
    // setupPointers ();
}

void reshape (int w, int h)
{
    glViewport (0, 0, (GLsizei) w, (GLsizei) h);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluOrtho2D (0.0, (GLdouble) w, 0.0, (GLdouble) h);
}

void mouse (int button, int state, int x, int y)
{
    switch (button) {
        case GLUT_LEFT_BUTTON:
            if (state == GLUT_DOWN) {
                if (derefMethod == DRAWARRAY)
                    derefMethod = ARRAYPELEMENT;
                else if (derefMethod == ARRAYPELEMENT)
                    derefMethod = DRAWELEMENTS;
                glutPostRedisplay();
            }
            break;
        default:
            break;
    }
}

```

```
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize (400, 400);
    glutInitWindowPosition (100, 100);
    glutCreateWindow (argv[0]);
    init ();
    glutDisplayFunc(display);
    glutReshapeFunc(reshape);
    glutMouseFunc(mouse);
    glutMainLoop();
    return 0;
}
```

Report:

The first part of the lab is not really hard. I changed the values of the triangles and made them look different. I replaced `glDrawArrays` and `glDrawElements` with `glArrayElements`. The extra credit problem is kind of tricky because once the front of back face is culled, it doesn't come back. I haven't figured out a way to fix it, but I'll work on it. Overall, I think I did good on this lab.