

## LAB SET 9

Write a program to fill a 3D cube with a set of six colors for its six faces with the Z-buffer hidden surface removal algorithm. Allow the user to view all faces of the cube with rotation.

```
//Lab Set 9 Program
//Akarsh Singh

#include <GL/glut.h>
double rotate_y=0;
double rotate_x=0;
void display()
{
    // Clear screen and Z-buffer
    glClear(GL_COLOR_BUFFER_BIT|GL_DEPTH_BUFFER_BIT);

    // Reset transformations
    glLoadIdentity(); // Rotate when user changes rotate_x and rotate_y
    glRotatef( rotate_x, 1.0, 0.0, 0.0 );
    glRotatef( rotate_y, 0.0, 1.0, 0.0 );

    //Multi-colored side - FRONT
    glBegin(GL_POLYGON);
    glColor3f( 1.0, 0.0, 0.0 );
    glVertex3f( 0.5, -0.5, -0.5 ); // P1 is red
    glColor3f( 0.0, 1.0, 0.0 );
    glVertex3f( 0.5, 0.5, -0.5 ); // P2 is green
    glColor3f( 0.0, 0.0, 1.0 );
    glVertex3f( -0.5, 0.5, -0.5 ); // P3 is blue
    glColor3f( 1.0, 0.0, 1.0 );
    glVertex3f( -0.5, -0.5, -0.5 ); // P4 is purple
    glEnd();

    // White side - BACK
    glBegin(GL_POLYGON);
    glColor3f( 1.0, 1.0, 1.0 );
    glVertex3f( 0.5, -0.5, 0.5 );
    glVertex3f( 0.5, 0.5, 0.5 );
    glVertex3f( -0.5, 0.5, 0.5 );
    glVertex3f( -0.5, -0.5, 0.5 );
    glEnd();

    // Purple side - RIGHT
    glBegin(GL_POLYGON);
    glColor3f( 1.0, 0.0, 1.0 );
    glVertex3f( 0.5, -0.5, -0.5 );
    glVertex3f( 0.5, 0.5, -0.5 );
    glVertex3f( 0.5, 0.5, 0.5 );
    glVertex3f( 0.5, -0.5, 0.5 );
    glEnd();
}
```

```

    glVertex3f( 0.5, -0.5, 0.5 );
    glEnd();
    // Green side - LEFT
    glBegin(GL_POLYGON);
    glColor3f( 0.0, 1.0, 0.0 );
    glVertex3f( -0.5, -0.5, 0.5 );
    glVertex3f( -0.5, 0.5, 0.5 );
    glVertex3f( -0.5, 0.5, -0.5 );
    glVertex3f( -0.5, -0.5, -0.5 );
    glEnd();

    // Blue side - TOP
    glBegin(GL_POLYGON);
    glColor3f( 0.0, 0.0, 1.0 );
    glVertex3f( 0.5, 0.5, 0.5 );
    glVertex3f( 0.5, 0.5, -0.5 );
    glVertex3f( -0.5, 0.5, -0.5 );
    glVertex3f( -0.5, 0.5, 0.5 );
    glEnd();

    // Red side - BOTTOM
    glBegin(GL_POLYGON);
    glColor3f( 1.0, 0.0, 0.0 );
    glVertex3f( 0.5, -0.5, -0.5 );
    glVertex3f( 0.5, -0.5, 0.5 );
    glVertex3f( -0.5, -0.5, 0.5 );
    glVertex3f( -0.5, -0.5, -0.5 );
    glEnd();
    glFlush();
    glutSwapBuffers();
}

// specialKeys() Callback Function
void specialKeys( int key, int x, int y )
{
    // Right arrow - increase rotation by 5 degree
    if (key == GLUT_KEY_RIGHT)
        rotate_y += 5;

    // Left arrow - decrease rotation by 5 degree
    else if (key == GLUT_KEY_LEFT)
        rotate_y -= 5;

    else if (key == GLUT_KEY_UP)
        rotate_x += 5;

    else if (key == GLUT_KEY_DOWN)
        rotate_x -= 5;

    // Request display update
    glutPostRedisplay();
}

```

```

}
int main(int argc, char* argv[])
{
    // Initialize GLUT and process user parameters
    glutInit(&argc,argv);

    // Request double buffered true color window with Z-buffer
    glutInitDisplayMode(GLUT_DOUBLE | GLUT_RGB | GLUT_DEPTH);

    // Create window
    glutCreateWindow("Rotating Cube");

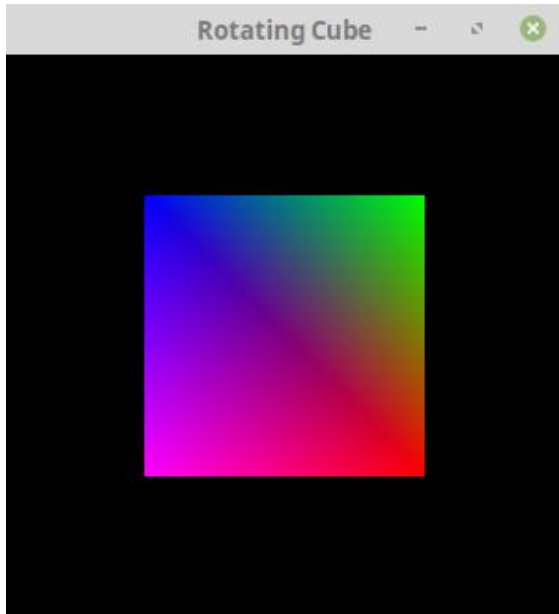
    // Enable Z-buffer depth test
    glEnable(GL_DEPTH_TEST); // Callback functions
    glutDisplayFunc(display);
    glutSpecialFunc(specialKeys);

    // Pass control to GLUT for events
    glutMainLoop();

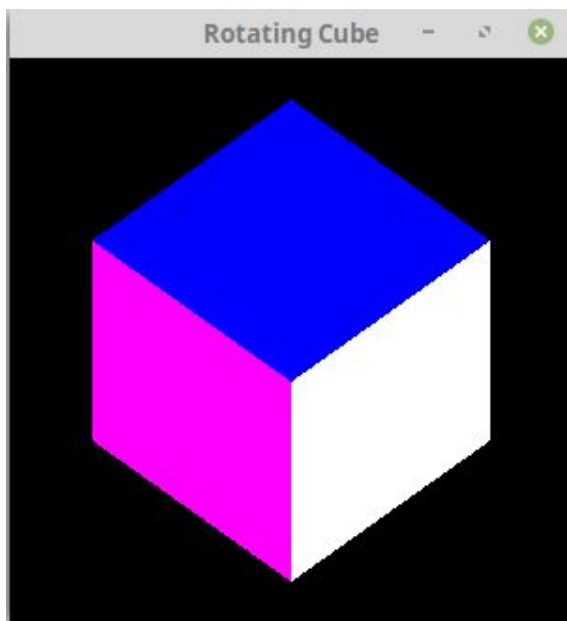
    return 0;
}

```

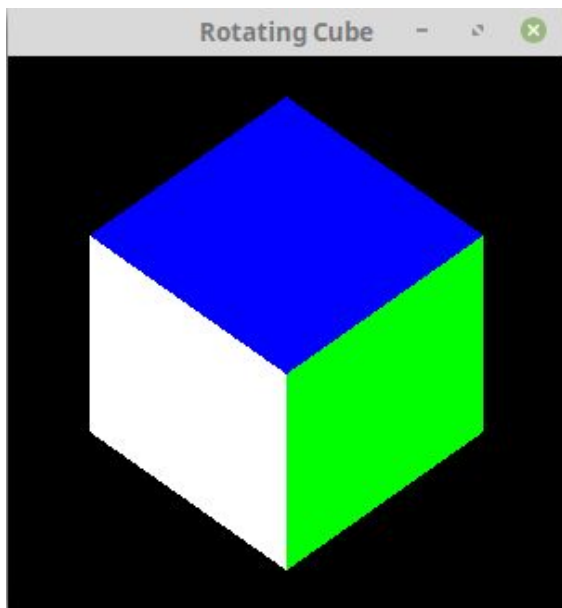
## OUTPUT



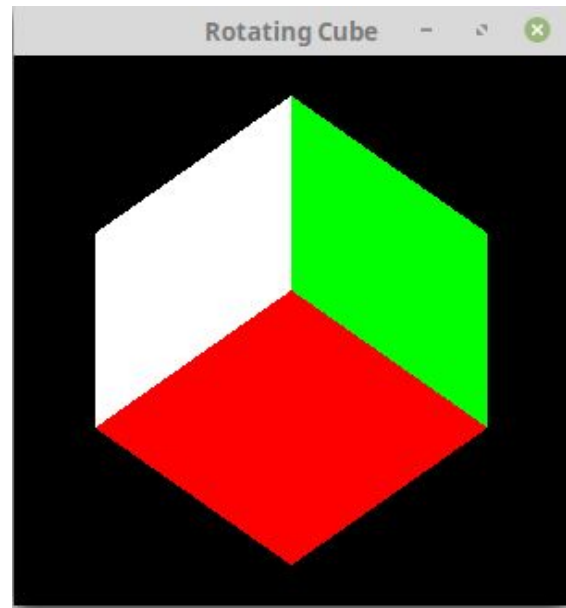
**Screenshot 1**



**Screenshot 2**



**Screenshot 3**



**Screenshot 4**