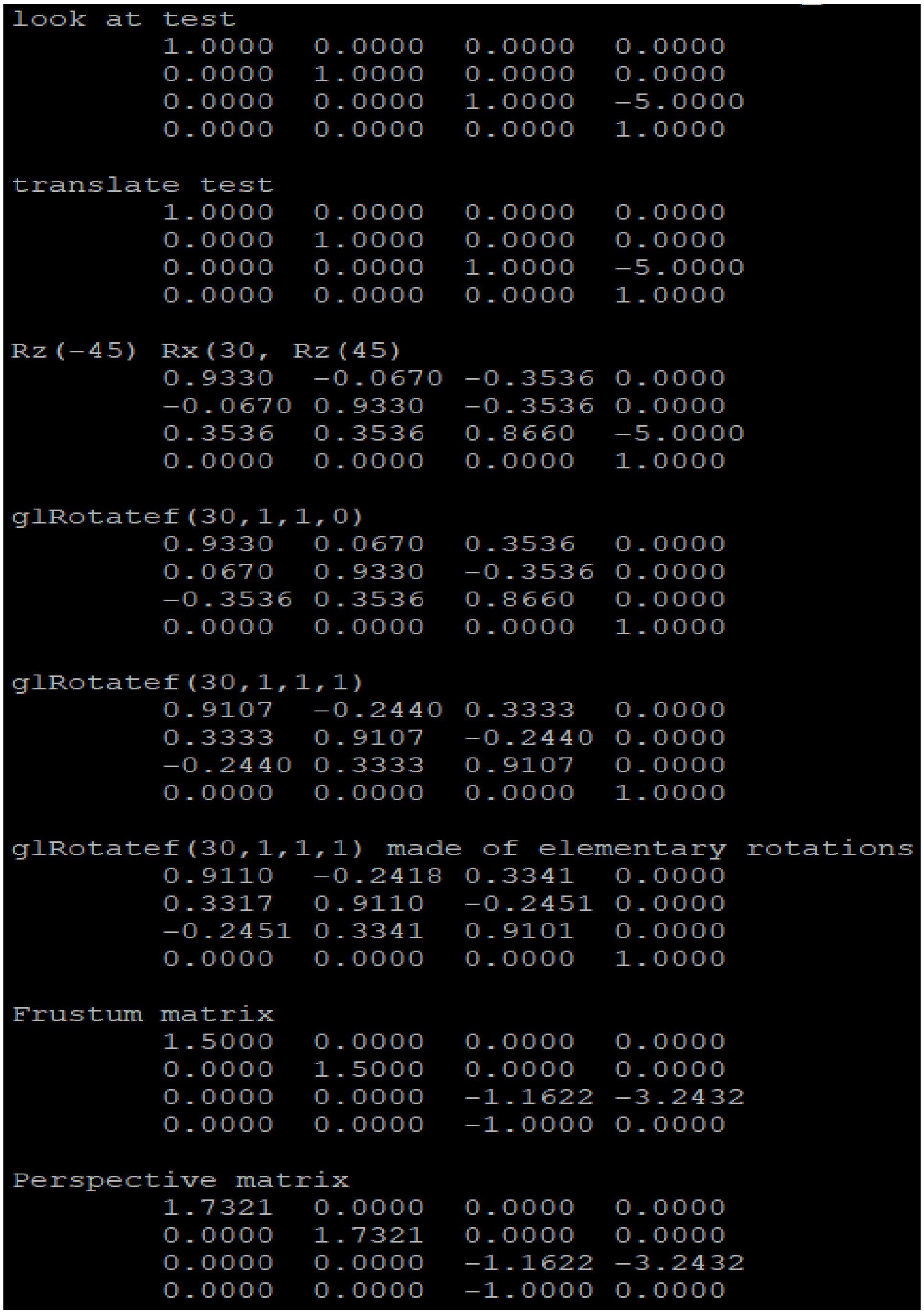
CSE 4200 Lab 10 – Spencer Wallace

Summary:

All parts completed successfully. Because I was able to complete all parts of the lab I am giving myself a full score of 30/30. For the look at vs translate test we do get the same outputs because the transformations are the same with respect to the camera. For the elementary rotation vs glRotatef(30,1,1,0) test we get outputs with the same values if we ignore the sign, but the outputs have different signs because of the different axis rotations.

Outputs on next page



```
look at test
        1.0000  0.0000  0.0000  0.0000
        0.0000  1.0000  0.0000  0.0000
        0.0000  0.0000  1.0000  -5.0000
        0.0000  0.0000  0.0000  1.0000

translate test
        1.0000  0.0000  0.0000  0.0000
        0.0000  1.0000  0.0000  0.0000
        0.0000  0.0000  1.0000  -5.0000
        0.0000  0.0000  0.0000  1.0000

Rz(-45) Rx(30, Rz(45)
        0.9330  -0.0670 -0.3536 0.0000
        -0.0670 0.9330  -0.3536 0.0000
        0.3536  0.3536  0.8660  -5.0000
        0.0000  0.0000  0.0000  1.0000

glRotatef(30,1,1,0)
        0.9330  0.0670  0.3536  0.0000
        0.0670  0.9330  -0.3536 0.0000
        -0.3536 0.3536  0.8660  0.0000
        0.0000  0.0000  0.0000  1.0000

glRotatef(30,1,1,1)
        0.9107  -0.2440 0.3333  0.0000
        0.3333  0.9107  -0.2440 0.0000
        -0.2440 0.3333  0.9107  0.0000
        0.0000  0.0000  0.0000  1.0000

glRotatef(30,1,1,1) made of elementary rotations
        0.9110  -0.2418 0.3341  0.0000
        0.3317  0.9110  -0.2451 0.0000
        -0.2451 0.3341  0.9101  0.0000
        0.0000  0.0000  0.0000  1.0000

Frustum matrix
        1.5000  0.0000  0.0000  0.0000
        0.0000  1.5000  0.0000  0.0000
        0.0000  0.0000  -1.1622 -3.2432
        0.0000  0.0000  -1.0000 0.0000

Perspective matrix
        1.7321  0.0000  0.0000  0.0000
        0.0000  1.7321  0.0000  0.0000
        0.0000  0.0000  -1.1622 -3.2432
        0.0000  0.0000  -1.0000 0.0000
```

Code starts on next page

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

using namespace std;

void init(void)
{
   glClearColor (1.0, 1.0, 1.0, 0.0);
   glShadeModel (GL_FLAT);
}

//print the transformation matrix
template<class T>
void print_mat(T m[][4])
{
  cout.precision(4);
  cout << fixed;
  for (int i = 0; i < 4; ++i) {
    cout << "\t";
    for (int j = 0; j < 4; ++j)
      cout << m[j][i] << "\t";
    cout << endl;
  }
  cout << endl;
```

```
}

void reshape(int w, int h)
{
  glViewport(0, 0, (GLsizei)w, (GLsizei)h);
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  glFrustum(-1.0, 1.0, -1.0, 1.0, 1.5, 20.0);
  glMatrixMode(GL_MODELVIEW);
}

void display(void)
{
  float p[4][4];
  double pd[4][4];
  double pdc[4][4];
  glClear(GL_COLOR_BUFFER_BIT);
  glColor3f(0.0, 0.0, 0.0); //black color
  glLoadIdentity(); // clear the matrix


  gluLookAt (0.0, 0.0, 5.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
 // glTranslatef( 0, 0, -5);
  glGetDoublev(GL_MODELVIEW_MATRIX, &pdc[0][0]);
  cout << "look at test" << endl;
  print_mat(pdc);

  glLoadIdentity();
  glTranslatef( 0, 0, -5);
  glGetDoublev(GL_MODELVIEW_MATRIX, &pdc[0][0]);
  cout << "translate test" << endl;
  print_mat(pdc);



  //glScalef (1.0, 2.0, 1.0);
  //glutWireCube (1.0);

  glRotatef(-45,0,0,1);
  glRotatef(30,1,0,0);
  glRotatef(45,0,0,1);

  glGetDoublev(GL_MODELVIEW_MATRIX, &pdc[0][0]);
  cout << "Rz(-45) Rx(30, Rz(45)" << endl;
  print_mat(pdc);

  glLoadIdentity();

  glRotatef(30,1,1,0);

  glGetDoublev(GL_MODELVIEW_MATRIX, &pdc[0][0]);
  cout << "glRotatef(30,1,1,0)" << endl;
  print_mat(pdc);


  glLoadIdentity();
  glRotatef(30,1,1,1);

  glGetDoublev(GL_MODELVIEW_MATRIX, &pdc[0][0]);
  cout << "glRotatef(30,1,1,1)" << endl;
  print_mat(pdc);

  glLoadIdentity();
  glRotatef(45,0,0,1);
  glRotatef(-35,0,1,0);
  glRotatef(30,1,0,0);
  glRotatef(35,0,1,0);
  glRotatef(-45,0,0,1);



  glGetDoublev(GL_MODELVIEW_MATRIX, &pdc[0][0]);
  cout << "glRotatef(30,1,1,1) made of elementary rotations" << endl;
  print_mat(pdc);

  glLoadIdentity();
  glFrustum(-1.0, 1.0, -1.0, 1.0, 1.5, 20.0);
  glGetDoublev(GL_MODELVIEW_MATRIX, &pdc[0][0]);
  cout << "Frustum matrix" << endl;
  print_mat(pdc);

  glLoadIdentity();
  gluPerspective( 60.0, 1.0, 1.5, 20.0 );
  glGetDoublev(GL_MODELVIEW_MATRIX, &pdc[0][0]);
  cout << "Perspective matrix" << endl;
  print_mat(pdc);
```

```
  glFlush();
}

void keyboard(unsigned char key, int x, int y)
{
  switch (key) {
  case 27:
    exit(0);
    break;
  }
}

int main(int argc, char** argv)
{
  glutInit(&argc, argv);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
  glutInitWindowSize(500, 500);
  glutInitWindowPosition(100, 100);
  glutCreateWindow(argv[0]);
  init();
  glutDisplayFunc(display);
  glutReshapeFunc(reshape);
  glutKeyboardFunc(keyboard);
  glutMainLoop();
  return 0;
}
```