**Lab Taks-3**

Submission Guidelines-

* Rename the file with your serial number only
* Must submit within time that will be discussed in class VUES
* Must include resources for all the section in the table

|  |
| --- |
| **Question- 1**  Draw five storied building with windows and a front door |
| **Graph Plot (Picture)-** |
| Code-  #include <windows.h>  #include <GL/glut.h>  void initGL() {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color  glBegin(GL\_QUADS); //main building  glColor3f(1.0f, 1.0f, 1.0f); // white  glVertex2f(-4,-2);  glVertex2f(-4,3);  glVertex2f(-2,3);  glVertex2f(-2,-2);  glEnd();  glBegin(GL\_LINES); //5th floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,-3); // x, y  glVertex2f(-2,-3); // x, y  glEnd();  glBegin(GL\_LINES); //4th floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,2); // x, y  glVertex2f(-2,2); // x, y  glEnd();  glBegin(GL\_LINES); //3rd floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,1); // x, y  glVertex2f(-2,1); // x, y  glEnd();  glBegin(GL\_LINES); //2nd floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,0); // x, y  glVertex2f(-2,0); // x, y  glEnd();  glBegin(GL\_LINES); //1st floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,-1); // x, y  glVertex2f(-2,-1); // x, y  glEnd();  glBegin(GL\_POLYGON); //5th floor window  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,2.4);  glVertex2f(-3.6,2.8);  glVertex2f(-3.2,2.8);  glVertex2f(-3.2,2.4);  glEnd();  glBegin(GL\_POLYGON); //5th floor window left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,2.4);  glVertex2f(-3.6,2.8);  glVertex2f(-3.2,2.8);  glVertex2f(-3.2,2.4);  glEnd();  glBegin(GL\_POLYGON); //5th floor window right  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,2.4);  glVertex2f(-2.6,2.8);  glVertex2f(-2.2,2.8);  glVertex2f(-2.2,2.4);  glEnd();  glBegin(GL\_POLYGON); //4th floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,1.4);  glVertex2f(-3.6,1.8);  glVertex2f(-3.2,1.8);  glVertex2f(-3.2,1.4);  glEnd();  glBegin(GL\_POLYGON); //4th floor window Right  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,1.4);  glVertex2f(-2.6,1.8);  glVertex2f(-2.2,1.8);  glVertex2f(-2.2,1.4);  glEnd();  glBegin(GL\_POLYGON); //3rd floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,0.4);  glVertex2f(-3.6,0.8);  glVertex2f(-3.2,0.8);  glVertex2f(-3.2,0.4);  glEnd();  glBegin(GL\_POLYGON); //3rd floor window Right  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,0.4);  glVertex2f(-2.6,0.8);  glVertex2f(-2.2,0.8);  glVertex2f(-2.2,0.4);  glEnd();  glBegin(GL\_POLYGON); //2nd floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,0.6);  glVertex2f(-3.6,0.2);  glVertex2f(-3.2,0.2);  glVertex2f(-3.2,0.6);  glEnd();  glBegin(GL\_POLYGON); //2nd floor window right  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,0.6);  glVertex2f(-2.6,0.2);  glVertex2f(-2.2,0.2);  glVertex2f(-2.2,0.6);  glEnd();  glBegin(GL\_POLYGON); //2nd floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,-0.6);  glVertex2f(-3.6,-0.2);  glVertex2f(-3.2,-0.2);  glVertex2f(-3.2,-0.6);  glEnd();  glBegin(GL\_POLYGON); //2nd floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,-0.6);  glVertex2f(-2.6,-0.2);  glVertex2f(-2.2,-0.2);  glVertex2f(-2.2,-0.6);  glEnd();  glBegin(GL\_POLYGON); //Ground floor Door  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.2,-2);  glVertex2f(-3.2,-1.2);  glVertex2f(-2.6,-1.2);  glVertex2f(-2.6,-2);  glEnd();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title  glutInitWindowSize(320, 320); // Set the window's initial width & height  glutInitWindowPosition(50, 50); // Position the window's initial top-left corner  gluOrtho2D(-5,5,-5,5);  glLineWidth(3.0f);  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw a tree |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  void initGL() {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color  glBegin(GL\_POLYGON); // tree base  glColor3ub(102,51,0); // brown  glVertex2f(-0.2,-2);  glVertex2f(-0.2,1);  glVertex2f(0.2, 1);  glVertex2f(0.2,-2);  glEnd();  glBegin(GL\_POLYGON); // tree base  glColor3ub(0,255,0); // green  glVertex2f(-1,2);  glVertex2f(0,3);  glVertex2f(1,2);  glEnd();  glBegin(GL\_POLYGON); // tree base  glColor3ub(0,255,0); //green  glVertex2f(-1,1.5);  glVertex2f(0,2.5);  glVertex2f(1,1.5);  glEnd();  glBegin(GL\_POLYGON); // tree base  glColor3ub(0,255,0); // green  glVertex2f(-1,1);  glVertex2f(0,2);  glVertex2f(1,1);  glEnd();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title  glutInitWindowSize(320, 320); // Set the window's initial width & height  glutInitWindowPosition(50, 50); // Position the window's initial top-left corner  gluOrtho2D(-5,5,-5,5);  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 3**  Draw a lamppost with black background |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  void initGL() {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color  glBegin(GL\_POLYGON); // lamp stand  glColor3ub(160,160,160); // black  glVertex2f(3,-2);  glVertex2f(3,1);  glVertex2f(3.5,1);  glVertex2f(3.5,-2);  glEnd();  glBegin(GL\_POLYGON); // lamp stand  glColor3ub(160,160,160); // black  glVertex2f(2,0.5);  glVertex2f(2,1);  glVertex2f(3.5,1);  glVertex2f(3.5,0.5);  glEnd();  glBegin(GL\_POLYGON); // light  glColor3ub(255,255,0); // black  glVertex2f(1.7,-0.1);  glVertex2f(1.7,0.3);  glVertex2f(2,0.5);  glVertex2f(2.3,0.3);  glVertex2f(2.3,-0.1);  glEnd();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title  glutInitWindowSize(320, 320); // Set the window's initial width & height  glutInitWindowPosition(50, 50); // Position the window's initial top-left corner  gluOrtho2D(-5,5,-5,5);  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 4**  Draw a bench |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  void initGL() {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Black and opaque  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color  glBegin(GL\_POLYGON); // bench leg  glColor3ub(204, 102, 0);  glVertex2f(4.4,-2);  glVertex2f(4.4,-1.6);  glVertex2f(4.6,-1.6);  glVertex2f(4.6,-2);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(5.8,-2);  glVertex2f(5.8,-1.6);  glVertex2f(6,-1.6);  glVertex2f(6,-2);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(4.8,-1.8);  glVertex2f(4.8,-1.6);  glVertex2f(5,-1.6);  glVertex2f(5,-1.8);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(5.4,-1.8);  glVertex2f(5.4,-1.6);  glVertex2f(5.6,-1.6);  glVertex2f(5.6,-1.8);  glEnd();  glBegin(GL\_POLYGON); //bench seat  glColor3ub(204, 102, 0);  glVertex2f(4,-1.6);  glVertex2f(4.2,-1.4);  glVertex2f(6.2,-1.4);  glVertex2f(6.4,-1.6);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(4.6,-1.4);  glVertex2f(4.6,-1.2);  glVertex2f(4.8,-1.2);  glVertex2f(4.8,-1.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(5.6,-1.4);  glVertex2f(5.6,-1.2);  glVertex2f(5.8,-1.2);  glVertex2f(5.8,-1.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(4.2,-1.2);  glVertex2f(4.2,-1);  glVertex2f(6.2,-1);  glVertex2f(6.2,-1.2);  glEnd();  glFlush(); // Render now  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title  glutInitWindowSize(320, 320); // Set the window's initial width & height  glutInitWindowPosition(50, 50); // Position the window's initial top-left corner  gluOrtho2D(-2,8,-4,3);  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 5**  Use the building, tree, lamppost and bench to create a scenario |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h>  #include <GL/glut.h>  void initGL() {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer with current clearing color  //------------------------------------------Building----------------------------------------------------------  glBegin(GL\_QUADS); //main building  glColor3f(1.0f, 1.0f, 1.0f); // white  glVertex2f(-4,-2);  glVertex2f(-4,3);  glVertex2f(-2,3);  glVertex2f(-2,-2);  glEnd();  glBegin(GL\_LINES); //5th floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,-3); // x, y  glVertex2f(-2,-3); // x, y  glEnd();  glBegin(GL\_LINES); //4th floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,2); // x, y  glVertex2f(-2,2); // x, y  glEnd();  glBegin(GL\_LINES); //3rd floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,1); // x, y  glVertex2f(-2,1); // x, y  glEnd();  glBegin(GL\_LINES); //2nd floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,0); // x, y  glVertex2f(-2,0); // x, y  glEnd();  glBegin(GL\_LINES); //1st floor line  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-4,-1); // x, y  glVertex2f(-2,-1); // x, y  glEnd();  glBegin(GL\_POLYGON); //5th floor window  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,2.4);  glVertex2f(-3.6,2.8);  glVertex2f(-3.2,2.8);  glVertex2f(-3.2,2.4);  glEnd();  glBegin(GL\_POLYGON); //5th floor window left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,2.4);  glVertex2f(-3.6,2.8);  glVertex2f(-3.2,2.8);  glVertex2f(-3.2,2.4);  glEnd();  glBegin(GL\_POLYGON); //5th floor window right  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,2.4);  glVertex2f(-2.6,2.8);  glVertex2f(-2.2,2.8);  glVertex2f(-2.2,2.4);  glEnd();  glBegin(GL\_POLYGON); //4th floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,1.4);  glVertex2f(-3.6,1.8);  glVertex2f(-3.2,1.8);  glVertex2f(-3.2,1.4);  glEnd();  glBegin(GL\_POLYGON); //4th floor window Right  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,1.4);  glVertex2f(-2.6,1.8);  glVertex2f(-2.2,1.8);  glVertex2f(-2.2,1.4);  glEnd();  glBegin(GL\_POLYGON); //3rd floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,0.4);  glVertex2f(-3.6,0.8);  glVertex2f(-3.2,0.8);  glVertex2f(-3.2,0.4);  glEnd();  glBegin(GL\_POLYGON); //3rd floor window Right  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,0.4);  glVertex2f(-2.6,0.8);  glVertex2f(-2.2,0.8);  glVertex2f(-2.2,0.4);  glEnd();  glBegin(GL\_POLYGON); //2nd floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,0.6);  glVertex2f(-3.6,0.2);  glVertex2f(-3.2,0.2);  glVertex2f(-3.2,0.6);  glEnd();  glBegin(GL\_POLYGON); //2nd floor window right  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,0.6);  glVertex2f(-2.6,0.2);  glVertex2f(-2.2,0.2);  glVertex2f(-2.2,0.6);  glEnd();  glBegin(GL\_POLYGON); //2nd floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.6,-0.6);  glVertex2f(-3.6,-0.2);  glVertex2f(-3.2,-0.2);  glVertex2f(-3.2,-0.6);  glEnd();  glBegin(GL\_POLYGON); //2nd floor window Left  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-2.6,-0.6);  glVertex2f(-2.6,-0.2);  glVertex2f(-2.2,-0.2);  glVertex2f(-2.2,-0.6);  glEnd();  glBegin(GL\_POLYGON); //Ground floor Door  glColor3f(0.0f, 0.0f, 0.0f); // black  glVertex2f(-3.2,-2);  glVertex2f(-3.2,-1.2);  glVertex2f(-2.6,-1.2);  glVertex2f(-2.6,-2);  glEnd();  //----------------------------------------------------------Tree--------------------------------  glBegin(GL\_POLYGON); // tree base  glColor3ub(102,51,0); // brown  glVertex2f(-0.2,-2);  glVertex2f(-0.2,1);  glVertex2f(0.2, 1);  glVertex2f(0.2,-2);  glEnd();  glBegin(GL\_POLYGON); // tree base  glColor3ub(0,255,0); // green  glVertex2f(-1,2);  glVertex2f(0,3);  glVertex2f(1,2);  glEnd();  glBegin(GL\_POLYGON); // tree base  glColor3ub(0,255,0); //green  glVertex2f(-1,1.5);  glVertex2f(0,2.5);  glVertex2f(1,1.5);  glEnd();  glBegin(GL\_POLYGON); // tree base  glColor3ub(0,255,0); // green  glVertex2f(-1,1);  glVertex2f(0,2);  glVertex2f(1,1);  glEnd();  //-------------------------------------------------Lamp Stand---------------------------------------  glBegin(GL\_POLYGON); // lamp stand  glColor3ub(160,160,160); // black  glVertex2f(3,-2);  glVertex2f(3,1);  glVertex2f(3.5,1);  glVertex2f(3.5,-2);  glEnd();  glBegin(GL\_POLYGON); // lamp stand  glColor3ub(160,160,160); // black  glVertex2f(2,0.5);  glVertex2f(2,1);  glVertex2f(3.5,1);  glVertex2f(3.5,0.5);  glEnd();  glBegin(GL\_POLYGON); // light  glColor3ub(255,255,0);  glVertex2f(1.7,-0.1);  glVertex2f(1.7,0.3);  glVertex2f(2,0.5);  glVertex2f(2.3,0.3);  glVertex2f(2.3,-0.1);  glEnd();  //----------------------------------------------------Bench----------------------------------------  glBegin(GL\_POLYGON); // bench leg  glColor3ub(204, 102, 0);  glVertex2f(4.4,-2);  glVertex2f(4.4,-1.6);  glVertex2f(4.6,-1.6);  glVertex2f(4.6,-2);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(5.8,-2);  glVertex2f(5.8,-1.6);  glVertex2f(6,-1.6);  glVertex2f(6,-2);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(4.8,-1.8);  glVertex2f(4.8,-1.6);  glVertex2f(5,-1.6);  glVertex2f(5,-1.8);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(5.4,-1.8);  glVertex2f(5.4,-1.6);  glVertex2f(5.6,-1.6);  glVertex2f(5.6,-1.8);  glEnd();  glBegin(GL\_POLYGON); //bench seat  glColor3ub(204, 102, 0);  glVertex2f(4,-1.6);  glVertex2f(4.2,-1.4);  glVertex2f(6.2,-1.4);  glVertex2f(6.4,-1.6);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(4.6,-1.4);  glVertex2f(4.6,-1.2);  glVertex2f(4.8,-1.2);  glVertex2f(4.8,-1.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(5.6,-1.4);  glVertex2f(5.6,-1.2);  glVertex2f(5.8,-1.2);  glVertex2f(5.8,-1.4);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(204, 102, 0);  glVertex2f(4.2,-1.2);  glVertex2f(4.2,-1);  glVertex2f(6.2,-1);  glVertex2f(6.2,-1.2);  glEnd();  glFlush(); // Render now  }  /\* Main function: GLUT runs as a console application starting at main() \*/  int main(int argc, char\*\* argv) {  glutInit(&argc, argv); // Initialize GLUT  glutCreateWindow("Vertex, Primitive & Color"); // Create window with the given title  glutInitWindowSize(320, 320); // Set the window's initial width & height  glutInitWindowPosition(50, 50); // Position the window's initial top-left corner  gluOrtho2D(-5,7,-5,7);  glLineWidth(3.0f);  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |