**Lab Taks-3**

Submission Guidelines-

* Rename the file with your serial number only
* Must submit within time that will be discussed in class in 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> // for MS Windows**  **#include <GL/glut.h>**  **#include <cmath> // GLUT, include glu.h and gl.h**  **void TOP()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(52, 73, 94);**  **glVertex2d(-4,6);**  **glVertex2d(0,8);**  **glVertex2d(4,6);**  **glEnd();**  **}**  **void Rectangle()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(153, 163, 164);**  **glVertex2d(-2,6);**  **glVertex2d(2,6);**  **glVertex2d(2,-4);**  **glVertex2d(-2,-4);**  **glEnd();**  **}**  **void BuildingLine()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,6);**  **glVertex2d(2,6);**  **glVertex2d(2,-4);**  **glVertex2d(-2,-4);**  **glEnd();**  **}**  **void Door()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.8237,-2.4);**  **glVertex2d(0.52269,-2.4);**  **glVertex2d(0.52269,-3.97944);**  **glVertex2d(-0.8237,-3.97944);**  **glEnd();**  **}**  **void DoorLine()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.8237,-2.4);**  **glVertex2d(0.52269,-2.4);**  **glVertex2d(0.52269,-3.97944);**  **glVertex2d(-0.8237,-3.97944);**  **glEnd();**  **}**  **void Line4()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,4);**  **glVertex2d(2,4);**  **glEnd();**  **}**  **void Line1()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,2);**  **glVertex2d(2,2);**  **glEnd();**  **}**  **void Line2()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,0);**  **glVertex2d(2,0);**  **glEnd();**  **}**  **void Line3()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,-2);**  **glVertex2d(2,-2);**  **glEnd();**  **}**  **void window1()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.56,3.36);**  **glVertex2d(0.63,3.36);**  **glVertex2d(0.63,2.57);**  **glVertex2d(-0.56,2.57);**  **glEnd();**  **}**  **void Windowline1(){**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.56,3.36);**  **glVertex2d(0.63,3.36);**  **glVertex2d(0.63,2.57);**  **glVertex2d(-0.56,2.57);**  **glEnd();**  **}**  **void windows1(){**  **window1();**  **Windowline1();**  **}**  **void window2()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.56,1.31);**  **glVertex2d(0.63,1.31);**  **glVertex2d(0.63,0.6);**  **glVertex2d(-0.56,0.6);**  **glEnd();**  **}**  **void Windowline2(){**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.56,1.31);**  **glVertex2d(0.63,1.31);**  **glVertex2d(0.63,0.6);**  **glVertex2d(-0.56,0.6);**  **glEnd();**  **}**  **void windows2(){**  **window2();**  **Windowline2();**  **}**  **void window3()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.56,5.24);**  **glVertex2d(0.63,5.24);**  **glVertex2d(0.63,4.5);**  **glVertex2d(-0.56,4.5);**  **glEnd();**  **}**  **void Windowline3(){**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.56,5.24);**  **glVertex2d(0.63,5.24);**  **glVertex2d(0.63,4.5);**  **glVertex2d(-0.56,4.5);**  **glEnd();**  **}**  **void windows3(){**  **window3();**  **Windowline3();**  **}**  **void window4()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.56,-0.62);**  **glVertex2d(0.63,-0.62);**  **glVertex2d(0.63,-1.35);**  **glVertex2d(-0.56,-1.35);**  **glEnd();**  **}**  **void Windowline4(){**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.56,-0.62);**  **glVertex2d(0.63,-0.62);**  **glVertex2d(0.63,-1.35);**  **glVertex2d(-0.56,-1.35);**  **glEnd();**  **}**  **void windows4(){**  **window4();**  **Windowline4();**  **}**  **void display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 0.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(1);**  **Rectangle();**  **BuildingLine();**  **Door();DoorLine();**  **Line1();Line2();Line3();Line4();**  **windows1();**  **windows2();**  **windows3();**  **windows4();**  **TOP();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(720, 720);**  **glutCreateWindow("Building"); // Create a window with the given title**  **// Set the window's initial width & height**  **glutDisplayFunc(display);**  **gluOrtho2D(-6,6,-6,8 );// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question- 2**  Draw a tree |
| **Graph Plot (Picture)-**  **A screenshot of a graphing graph  Description automatically generated** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h>**  **#include <cmath> // GLUT, include glu.h and gl.h**  **void Rectangle()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(110, 44, 0);**  **glVertex2d(-1.0,-1.0);**  **glVertex2d(0,-1.0);**  **glVertex2d(0,-3);**  **glVertex2d(-1,-3);**  **glEnd();**  **}**  **void Triangle1()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(30, 132, 73 );**  **glVertex2d(-3.0,-1.0);**  **glVertex2d(-0.46,0.48);**  **glVertex2d(2,-1);**  **glEnd();**  **}**  **void Triangle2()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(20, 90, 50 );**  **glVertex2d(-2.0,0);**  **glVertex2d(-0.48,1.48);**  **glVertex2d(1.0,0);**  **glEnd();**  **}**  **void display()**  **{**  **glClearColor(1.0f, 1.0f, 1.0f, 0.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(1.5);**  **Rectangle();**  **Triangle1();**  **Triangle2();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(720, 720);**  **glutCreateWindow("Tree"); // Create a window with the given title**  **// Set the window's initial width & height**  **glutDisplayFunc(display);**  **gluOrtho2D(-6,6,-6,6);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question- 3**  Draw a lamppost with black background |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h>**  **#include <cmath> // GLUT, include glu.h and gl.h**  **void TriangleofLamp(){**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 0 );**  **glVertex2d(-7.0,6.0);**  **glVertex2d(-6.0,6.0);**  **glVertex2d(-6.72044,5.3);**  **glEnd();**  **}**  **void BodyLamp(){**  **glBegin(GL\_POLYGON);**  **glColor3ub(52, 152, 219);**  **glVertex2d(-7.66,6.66);**  **glVertex2d(-6.0,6.66);**  **glVertex2d(-6.0,6.0);**  **glVertex2d(-7.0,6.0);**  **glVertex2d(-7.1,6.47);**  **glVertex2d(-7.68,6.47);**  **glVertex2d(-8.8,6.5);**  **glEnd();**  **}**  **void BottomLamp(){**  **glBegin(GL\_QUADS);**  **glColor3ub(52, 152, 219);**  **glVertex2d(-8.6,-2.0);**  **glVertex2d(-8.0,-2.0);**  **glVertex2d(-8.0,-2.45);**  **glVertex2d(-8.6,-2.45);**  **glEnd();**  **}**  **void LineLamp(){**  **glBegin(GL\_LINES);**  **glColor3ub(52, 152, 219);**  **glVertex2d(-8.2,6.5);**  **glVertex2d(-8.2,-2.0);**  **glEnd();**  **}**  **void display()**  **{**  **glClearColor(0.0f, 0.0f, 0.0f, 0.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(3);**  **TriangleofLamp();**  **BodyLamp();**  **BottomLamp();**  **LineLamp();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(720, 720);**  **glutCreateWindow("LampPOst"); // Create a window with the given title**  **// Set the window's initial width & height**  **glutDisplayFunc(display);**  **gluOrtho2D(-15,15,-15,15);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |

|  |
| --- |
| **Question- 4**  Draw a bench |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h>**  **#include <cmath> // GLUT, include glu.h and gl.h**  **void RectangleofBench()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(211, 84, 0 );**  **glVertex2d(9.0,-0.6);**  **glVertex2d(13.0,-0.6);**  **glVertex2d(13.0,-0.8);**  **glVertex2d(9.0,-0.8);**  **glEnd();**  **}**  **void leg1()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(0,0,0);**  **glVertex2d(9.8,-0.8);**  **glVertex2d(10.0,-0.8);**  **glVertex2d(10.0,-1.4);**  **glVertex2d(9.8,-1.4);**  **glEnd();**  **}**  **void leg2()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(0,0,0);**  **glVertex2d(11.8,-0.8);**  **glVertex2d(12.0,-0.8);**  **glVertex2d(12.0,-1.38);**  **glVertex2d(11.8,-1.38);**  **glEnd();**  **}**  **void display()**  **{**  **glClearColor(0.0f, 0.3f, 1.0f, 0.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(1);**  **RectangleofBench();**  **leg1();**  **leg2();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(720, 720);**  **glutCreateWindow("Bench"); // Create a window with the given title**  **// Set the window's initial width & height**  **glutDisplayFunc(display);**  **gluOrtho2D(8,14,-4,0);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A computer screen shot of a blue and black rectangular object  Description automatically generated** |

|  |
| --- |
| **Question- 5**  Use the building, tree, lamppost and bench to create a scenario |
| **Graph Plot (Picture)-**  A screenshot of a computer  Description automatically generated |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h>**  **#include <cmath> // GLUT, include glu.h and gl.h**  **void TOP()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(52, 73, 94);**  **glVertex2d(-4,6);**  **glVertex2d(0,8);**  **glVertex2d(4,6);**  **glEnd();**  **}**  **void Rectangle()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(153, 163, 164);**  **glVertex2d(-2,6);**  **glVertex2d(2,6);**  **glVertex2d(2,-4);**  **glVertex2d(-2,-4);**  **glEnd();**  **}**  **void BuildingLine()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,6);**  **glVertex2d(2,6);**  **glVertex2d(2,-4);**  **glVertex2d(-2,-4);**  **glEnd();**  **}**  **void Door()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.8237,-2.4);**  **glVertex2d(0.52269,-2.4);**  **glVertex2d(0.52269,-3.97944);**  **glVertex2d(-0.8237,-3.97944);**  **glEnd();**  **}**  **void DoorLine()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.8237,-2.4);**  **glVertex2d(0.52269,-2.4);**  **glVertex2d(0.52269,-3.97944);**  **glVertex2d(-0.8237,-3.97944);**  **glEnd();**  **}**  **void Line4()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,4);**  **glVertex2d(2,4);**  **glEnd();**  **}**  **void Line1()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,2);**  **glVertex2d(2,2);**  **glEnd();**  **}**  **void Line2()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,0);**  **glVertex2d(2,0);**  **glEnd();**  **}**  **void Line3()**  **{**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(0,0,0);**  **glVertex2d(-2,-2);**  **glVertex2d(2,-2);**  **glEnd();**  **}**  **void window1()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.56,3.36);**  **glVertex2d(0.63,3.36);**  **glVertex2d(0.63,2.57);**  **glVertex2d(-0.56,2.57);**  **glEnd();**  **}**  **void Windowline1(){**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.56,3.36);**  **glVertex2d(0.63,3.36);**  **glVertex2d(0.63,2.57);**  **glVertex2d(-0.56,2.57);**  **glEnd();**  **}**  **void windows1(){**  **window1();**  **Windowline1();**  **}**  **void window2()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.56,1.31);**  **glVertex2d(0.63,1.31);**  **glVertex2d(0.63,0.6);**  **glVertex2d(-0.56,0.6);**  **glEnd();**  **}**  **void Windowline2(){**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.56,1.31);**  **glVertex2d(0.63,1.31);**  **glVertex2d(0.63,0.6);**  **glVertex2d(-0.56,0.6);**  **glEnd();**  **}**  **void windows2(){**  **window2();**  **Windowline2();**  **}**  **void window3()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.56,5.24);**  **glVertex2d(0.63,5.24);**  **glVertex2d(0.63,4.5);**  **glVertex2d(-0.56,4.5);**  **glEnd();**  **}**  **void Windowline3(){**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.56,5.24);**  **glVertex2d(0.63,5.24);**  **glVertex2d(0.63,4.5);**  **glVertex2d(-0.56,4.5);**  **glEnd();**  **}**  **void windows3(){**  **window3();**  **Windowline3();**  **}**  **void window4()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(0,0,0);**  **glVertex2d(-0.56,-0.62);**  **glVertex2d(0.63,-0.62);**  **glVertex2d(0.63,-1.35);**  **glVertex2d(-0.56,-1.35);**  **glEnd();**  **}**  **void Windowline4(){**  **glBegin(GL\_LINE\_LOOP);**  **glColor3ub(255,255,255);**  **glVertex2d(-0.56,-0.62);**  **glVertex2d(0.63,-0.62);**  **glVertex2d(0.63,-1.35);**  **glVertex2d(-0.56,-1.35);**  **glEnd();**  **}**  **void windows4(){**  **window4();**  **Windowline4();**  **}**  **void displayBuilding()**  **{**  **glLineWidth(1);**  **Rectangle();**  **BuildingLine();**  **Door();DoorLine();**  **Line1();Line2();Line3();Line4();**  **windows1();**  **windows2();**  **windows3();**  **windows4();**  **TOP();**  **}**  **void RectangleofTree()**  **{**  **glBegin(GL\_QUADS);**  **glColor3ub(110, 44, 0);**  **glVertex2d(-1.0-6,-1.0);**  **glVertex2d(0-6,-1.0);**  **glVertex2d(0-6,-4);**  **glVertex2d(-1-6,-4);**  **glEnd();**  **}**  **void Triangle1()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(30, 132, 73 );**  **glVertex2d(-3.0-6,-1.0);**  **glVertex2d(-0.46-6,0.48);**  **glVertex2d(2-6,-1);**  **glEnd();**  **}**  **void Triangle2()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(20, 90, 50 );**  **glVertex2d(-2.0-6,0);**  **glVertex2d(-0.48-6,1.48);**  **glVertex2d(1.0-6,0);**  **glEnd();**  **}**  **void Treedisplay()**  **{**  **glLineWidth(1.5);**  **RectangleofTree();**  **Triangle1();**  **Triangle2();**  **}**  **void TriangleofLamp(){**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 0 );**  **glVertex2d(-7.0-2,6.0);**  **glVertex2d(-6.0-2,6.0);**  **glVertex2d(-6.72044-2,5.3);**  **glEnd();**  **}**  **void BodyLamp(){**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0 );**  **glVertex2d(-7.66-2,6.66);**  **glVertex2d(-6.0-2,6.66);**  **glVertex2d(-6.0-2,6.0);**  **glVertex2d(-7.0-2,6.0);**  **glVertex2d(-7.1-2,6.47);**  **glVertex2d(-7.68-2,6.47);**  **glVertex2d(-8.8-2,6.5);**  **glEnd();**  **}**  **void BottomLamp(){**  **glBegin(GL\_QUADS);**  **glColor3ub(0, 0, 0 );**  **glVertex2d(-8.6-2,-2.0);**  **glVertex2d(-8.0-2,-2.0);**  **glVertex2d(-8.0-2,-4);**  **glVertex2d(-8.6-2,-4);**  **glEnd();**  **}**  **void LineLamp(){**  **glBegin(GL\_LINES);**  **glColor3ub(0, 0, 0 );**  **glVertex2d(-8.2-2,6.5);**  **glVertex2d(-8.2-2,-2.0);**  **glEnd();**  **}**  **void LampDisplay(){**  **TriangleofLamp();**  **BodyLamp();**  **BottomLamp();**  **LineLamp();}**  **void RectangleofBench()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(211, 84, 0 );**  **glVertex2d(9.0-3.5,-0.6-2.6);**  **glVertex2d(13.0-3.5,-0.6-2.6);**  **glVertex2d(13.0-3.5,-0.8-2.6);**  **glVertex2d(9.0-3.5,-0.8-2.6);**  **glEnd();**  **}**  **void leg1()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(0,0,0);**  **glVertex2d(9.8-3.5,-0.8-2.6);**  **glVertex2d(10.0-3.5,-0.8-2.6);**  **glVertex2d(10.0-3.5,-1.4-2.6);**  **glVertex2d(9.8-3.5,-1.4-2.6);**  **glEnd();**  **}**  **void leg2()**  **{**  **glBegin(GL\_POLYGON);**  **glColor3ub(0,0,0);**  **glVertex2d(11.8-3.5,-0.8-2.6);**  **glVertex2d(12.0-3.5,-0.8-2.6);**  **glVertex2d(12.0-3.5,-1.4-2.6);**  **glVertex2d(11.8-3.5,-1.4-2.6);**  **glEnd();**  **}**  **void Benchdisplay()**  **{**  **RectangleofBench();**  **leg1();**  **leg2();**  **// Render now**  **}**  **void display()**  **{**  **glClearColor(0.0f, 0.3f, 1.0f, 0.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(1);**  **displayBuilding();**  **Treedisplay();**  **LampDisplay();**  **Benchdisplay();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv)**  **{**  **glutInit(&argc, argv); // Initialize GLUT**  **glutInitWindowSize(720, 720);**  **glutCreateWindow("Scenario"); // Create a window with the given title**  **// Set the window's initial width & height**  **glutDisplayFunc(display);**  **gluOrtho2D(-15,15,-15,15);// Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **A screenshot of a computer  Description automatically generated** |