**Lab Taks-3**

Submission Guidelines-

* Rename the file to your id only. If your id is 18-XXXXX-1, then the file name must be 18-XXXXX-1.docx.
* Must submit within time that will be discussed in class VUES to the section named Lab Tak-3
* 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(1.0f, 1.0f, 1.0f, 1.0f);  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  //  //floor1-2  //  glBegin(GL\_POLYGON);  glColor3f(0.659f, 0.831f, 0.49f);  glVertex2f(-4.0f, -4.0f);  glVertex2f(4.0f, -4.0f);  glVertex2f(4.0f, -4.0f);  glVertex2f(4.0f, 6.0f);  glVertex2f(4.0f, 6.0f);  glVertex2f(-4.0f, 6.0f);  glVertex2f(-4.0f, 6.0f);  glVertex2f(-4.0f, -4.0f);  glEnd();  //  //floor1-2--outline  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(-4.0f, -4.0f);  glVertex2f(4.0f, -4.0f);  glVertex2f(4.0f, -4.0f);  glVertex2f(4.0f, 6.0f);  glVertex2f(4.0f, 6.0f);  glVertex2f(-4.0f, 6.0f);  glVertex2f(-4.0f, 6.0f);  glVertex2f(-4.0f, -4.0f);  glEnd();  //  //win1  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(-3.0f, 2.0f);  glVertex2f(-1.0f, 2.0f);  glVertex2f(-1.0f, 2.0f);  glVertex2f(-1.0f, 4.0f);  glVertex2f(-1.0f, 4.0f);  glVertex2f(-3.0f, 4.0f);  glVertex2f(-3.0f, 4.0f);  glVertex2f(-3.0f, 2.0f);  glEnd();  //  //win2  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(1.0f, 2.0f);  glVertex2f(3.0f, 2.0f);  glVertex2f(3.0f, 2.0f);  glVertex2f(3.0f, 4.0f);  glVertex2f(3.0f, 4.0f);  glVertex2f(1.0f, 4.0f);  glVertex2f(1.0f, 4.0f);  glVertex2f(1.0f, 2.0f);  glEnd();  //  //door  //  glBegin(GL\_POLYGON);  glColor3f(0.9f, 0.6f, 0.0f);  glVertex2f(1.0f, -4.0f);  glVertex2f(3.0f, -4.0f);  glVertex2f(3.0f, -4.0f);  glVertex2f(3.0f, 0.0f);  glVertex2f(3.0f, 0.0f);  glVertex2f(1.0f, 0.0f);  glVertex2f(1.0f, 0.0f);  glVertex2f(1.0f, -4.0f);  glEnd();  //  //roof1  //  glBegin(GL\_POLYGON);  glColor3f(0.212f, 0.275f, 0.243f);  glVertex2f(-6.0f, 6.0f);  glVertex2f(6.0f, 6.0f);  glVertex2f(6.0f, 6.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(-4.0f, 8.0f);  glVertex2f(-4.0f, 8.0f);  glVertex2f(-6.0f, 6.0f);  glEnd();  //  //floor3-4  //  glBegin(GL\_POLYGON);  glColor3f(0.659f, 0.831f, 0.49f);  glVertex2f(-4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(-4.0f, 8.0f);  glEnd();  //  //floor 3-4-outline  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(-4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(-4.0f, 8.0f);  glEnd();  //  //win1  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(-3.0f, 14.0f);  glVertex2f(-1.0f, 14.0f);  glVertex2f(-1.0f, 14.0f);  glVertex2f(-1.0f, 16.0f);  glVertex2f(-1.0f, 16.0f);  glVertex2f(-3.0f, 16.0f);  glVertex2f(-3.0f, 16.0f);  glVertex2f(-3.0f, 14.0f);  glEnd();  //  //win2  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(1.0f, 14.0f);  glVertex2f(3.0f, 14.0f);  glVertex2f(3.0f, 14.0f);  glVertex2f(3.0f, 16.0f);  glVertex2f(3.0f, 16.0f);  glVertex2f(1.0f, 16.0f);  glVertex2f(1.0f, 16.0f);  glVertex2f(1.0f, 14.0f);  glEnd();  //  //win3  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(1.0f, 10.0f);  glVertex2f(3.0f, 10.0f);  glVertex2f(3.0f, 10.0f);  glVertex2f(3.0f, 12.0f);  glVertex2f(3.0f, 12.0f);  glVertex2f(1.0f, 12.0f);  glVertex2f(1.0f, 12.0f);  glVertex2f(1.0f, 10.0f);  glEnd();  //  //win4  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(-3.0f, 10.0f);  glVertex2f(-1.0f, 10.0f);  glVertex2f(-1.0f, 10.0f);  glVertex2f(-1.0f, 12.0f);  glVertex2f(-1.0f, 12.0f);  glVertex2f(-3.0f, 12.0f);  glVertex2f(-3.0f, 12.0f);  glVertex2f(-3.0f, 10.0f);  glEnd();  //  //main roof  //  glBegin(GL\_POLYGON);  glColor3f(0.212f, 0.275f, 0.243f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(0.0f, 22.0f);  glVertex2f(0.0f, 22.0f);  glVertex2f(-4.0f, 18.0f);  glEnd();  glFlush();  }  /\* 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(720, 720); // Set the window's initial width & height  glutInitWindowPosition(20, 20); // Position the window's initial top-left corner  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  gluOrtho2D(-20,20,-20,30);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw a tree |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h>  #include <GL/glut.h>  void initGL() {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  //  //tree  //  glBegin(GL\_POLYGON);  glColor3f(0.678f, 0.460f, 0.082f);  glVertex2f(-16.0f, -4.0f);  glVertex2f(-12.0f, -4.0f);  glVertex2f(-12.0f, -4.0f);  glVertex2f(-12.0f, 4.0f);  glVertex2f(-12.0f, 4.0f);  glVertex2f(-16.0f, 4.0f);  glVertex2f(-16.0f, 4.0f);  glVertex2f(-16.0f, -4.0f);  glEnd();  //  //1  //  glBegin(GL\_POLYGON);  glColor3f(0.043f, 0.769f, 0.141f);  glVertex2f(-20.0f, 2.0f);  glVertex2f(-8.0f, 2.0);  glVertex2f(-8.0f, 2.0);  glVertex2f(-14.0f, 8.0);  glVertex2f(-14.0f, 8.0);  glVertex2f(-20.0f, 2.0);  glEnd();  //  // 2  //  glBegin(GL\_POLYGON);  glColor3f(0.043f, 0.769f, 0.141f);  glVertex2f(-19.0f, 6.0);  glVertex2f(-9.0f, 6.0);  glVertex2f(-9.0f, 6.0);  glVertex2f(-14.0f, 12.0);  glVertex2f(-14.0f, 12.0);  glVertex2f(-19.0f, 6.0);  glEnd();  //  // 3  //  glBegin(GL\_POLYGON);  glColor3f(0.043f, 0.769f, 0.141f);  glVertex2f(-18.0f, 10.0);  glVertex2f(-10.0f, 10.0);  glVertex2f(-10.0f, 10.0);  glVertex2f(-14.0f, 14.0);  glVertex2f(-14.0f, 14.0);  glVertex2f(-18.0f, 10.0);  glEnd();  glFlush();  }  /\* 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(720, 720); // Set the window's initial width & height  glutInitWindowPosition(20, 20); // Position the window's initial top-left corner  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  gluOrtho2D(-50,20,-20,20);  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>  #include <GL/glut.h>  #include <math.h>  void initGL() {  glClearColor(0.0f, 0.0f, 0.0f, 0.0f);  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  //  //main stand  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(12.0f, -4.0f);  glVertex2f(13.0f, -4.0f);  glVertex2f(13.0f, -4.0f);  glVertex2f(13.0f, 10.0f);  glVertex2f(13.0f, 10.0f);  glVertex2f(12.0f, 10.0f);  glVertex2f(12.0f, 10.0f);  glVertex2f(12.0f, -4.0f);  glEnd();  //  //side stand  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(11.6f, 9.0f);  glVertex2f(17.4f, 9.0f);  glVertex2f(17.4f, 9.0f);  glVertex2f(17.4f, 9.4f);  glVertex2f(17.0f, 9.4f);  glVertex2f(11.6f, 9.4f);  glVertex2f(11.6f, 9.4f);  glVertex2f(11.6f, 9.0f);  glEnd();  //  //light stand  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(15.4f, 7.8f);  glVertex2f(17.0f, 7.8f);  glVertex2f(17.0f, 7.8f);  glVertex2f(16.2f, 9.0f);  glVertex2f(16.2f, 9.0f);  glVertex2f(15.4f, 7.8f);  glEnd();  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(255, 255, 0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=0.35;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+16.25, y+7.41);  }  glEnd();  glFlush();  }  /\* 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(720, 720); // Set the window's initial width & height  glutInitWindowPosition(20, 20); // Position the window's initial top-left corner  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  gluOrtho2D(-0,30,-15,20);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 4**  Draw a bench |
| **Graph Plot (Picture)-** |
| **Code-**  #include <windows.h>  #include <GL/glut.h>  void initGL() {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  //  //side stand-1  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(14.0f, -4.0f);  glVertex2f(14.2f, -4.0f);  glVertex2f(14.2f, -4.0f);  glVertex2f(14.2f, 1.0f);  glVertex2f(14.2f, 1.0f);  glVertex2f(14.0f, 1.0f);  glVertex2f(14.0f, 1.0f);  glVertex2f(14.0f, -4.0f);  glEnd();  //  //side stand-2  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(21.8f, -4.0f);  glVertex2f(22.0f, -4.0f);  glVertex2f(22.0f, -4.0f);  glVertex2f(22.0f, 1.0f);  glVertex2f(22.0f, 1.0f);  glVertex2f(21.8f, 1.0f);  glVertex2f(21.8f, 1.0f);  glVertex2f(21.8f, -4.0f);  glEnd();  //  //bench-1  //  glBegin(GL\_POLYGON);  glColor3f(0.9f, 0.6f, 0.0f);  glVertex2f(13.5f, -3.0f);  glVertex2f(22.5f, -3.0f);  glVertex2f(22.5f, -3.0f);  glVertex2f(22.5f, -2.5f);  glVertex2f(22.5f, -2.5f);  glVertex2f(13.5f, -2.5f);  glVertex2f(13.5f, -2.5f);  glVertex2f(13.5f, -3.0f);  glEnd();  //  //bench-1-outline  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(13.5f, -3.0f);  glVertex2f(22.5f, -3.0f);  glVertex2f(22.5f, -3.0f);  glVertex2f(22.5f, -2.5f);  glVertex2f(22.5f, -2.5f);  glVertex2f(13.5f, -2.5f);  glVertex2f(13.5f, -2.5f);  glVertex2f(13.5f, -3.0f);  glEnd();  //  //bench-2  //  glBegin(GL\_POLYGON);  glColor3f(0.9f, 0.6f, 0.0f);  glVertex2f(13.5f, -1.5f);  glVertex2f(22.5f, -1.5f);  glVertex2f(22.5f, -1.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(13.5f, -0.5f);  glVertex2f(13.5f, -1.5f);  glEnd();  //  //bench-2--outine  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(13.5f, -1.5f);  glVertex2f(22.5f, -1.5f);  glVertex2f(22.5f, -1.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(13.5f, -0.5f);  glVertex2f(13.5f, -1.5f);  glEnd();  //  //bench-3  //  glBegin(GL\_POLYGON);  glColor3f(0.9f, 0.6f, 0.0f);  glVertex2f(13.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, 0.5f);  glVertex2f(22.5f, 0.5f);  glVertex2f(13.5f, 0.5f);  glVertex2f(13.5f, 0.5f);  glVertex2f(13.5f, -0.5f);  glEnd();  //  //bench-3-outline  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(13.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, 0.5f);  glVertex2f(22.5f, 0.5f);  glVertex2f(13.5f, 0.5f);  glVertex2f(13.5f, 0.5f);  glVertex2f(13.5f, -0.5f);  glEnd();  glFlush();  }  /\* 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(720, 720); // Set the window's initial width & height  glutInitWindowPosition(20, 20); // Position the window's initial top-left corner  glutDisplayFunc(display); // Register callback handler for window re-paint event  initGL(); // Our own OpenGL initialization  gluOrtho2D(-0,30,-20,10);  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>  #include <math.h>  void tree() {  //  //tree  //  glBegin(GL\_POLYGON);  glColor3f(0.678f, 0.460f, 0.082f);  glVertex2f(-16.0f, -4.0f);  glVertex2f(-12.0f, -4.0f);  glVertex2f(-12.0f, -4.0f);  glVertex2f(-12.0f, 4.0f);  glVertex2f(-12.0f, 4.0f);  glVertex2f(-16.0f, 4.0f);  glVertex2f(-16.0f, 4.0f);  glVertex2f(-16.0f, -4.0f);  glEnd();  //  //1  //  glBegin(GL\_POLYGON);  glColor3f(0.043f, 0.769f, 0.141f);  glVertex2f(-20.0f, 2.0f);  glVertex2f(-8.0f, 2.0);  glVertex2f(-8.0f, 2.0);  glVertex2f(-14.0f, 8.0);  glVertex2f(-14.0f, 8.0);  glVertex2f(-20.0f, 2.0);  glEnd();  //  // 2  //  glBegin(GL\_POLYGON);  glColor3f(0.043f, 0.769f, 0.141f);  glVertex2f(-19.0f, 6.0);  glVertex2f(-9.0f, 6.0);  glVertex2f(-9.0f, 6.0);  glVertex2f(-14.0f, 12.0);  glVertex2f(-14.0f, 12.0);  glVertex2f(-19.0f, 6.0);  glEnd();  //  // 3  //  glBegin(GL\_POLYGON);  glColor3f(0.043f, 0.769f, 0.141f);  glVertex2f(-18.0f, 10.0);  glVertex2f(-10.0f, 10.0);  glVertex2f(-10.0f, 10.0);  glVertex2f(-14.0f, 14.0);  glVertex2f(-14.0f, 14.0);  glVertex2f(-18.0f, 10.0);  glEnd();  }  void building() {  //  //floor1-2  //  glBegin(GL\_POLYGON);  glColor3f(0.659f, 0.831f, 0.49f);  glVertex2f(-4.0f, -4.0f);  glVertex2f(4.0f, -4.0f);  glVertex2f(4.0f, -4.0f);  glVertex2f(4.0f, 6.0f);  glVertex2f(4.0f, 6.0f);  glVertex2f(-4.0f, 6.0f);  glVertex2f(-4.0f, 6.0f);  glVertex2f(-4.0f, -4.0f);  glEnd();  //  //floor1-2--outline  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(-4.0f, -4.0f);  glVertex2f(4.0f, -4.0f);  glVertex2f(4.0f, -4.0f);  glVertex2f(4.0f, 6.0f);  glVertex2f(4.0f, 6.0f);  glVertex2f(-4.0f, 6.0f);  glVertex2f(-4.0f, 6.0f);  glVertex2f(-4.0f, -4.0f);  glEnd();  //  //win1  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(-3.0f, 2.0f);  glVertex2f(-1.0f, 2.0f);  glVertex2f(-1.0f, 2.0f);  glVertex2f(-1.0f, 4.0f);  glVertex2f(-1.0f, 4.0f);  glVertex2f(-3.0f, 4.0f);  glVertex2f(-3.0f, 4.0f);  glVertex2f(-3.0f, 2.0f);  glEnd();  //  //win2  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(1.0f, 2.0f);  glVertex2f(3.0f, 2.0f);  glVertex2f(3.0f, 2.0f);  glVertex2f(3.0f, 4.0f);  glVertex2f(3.0f, 4.0f);  glVertex2f(1.0f, 4.0f);  glVertex2f(1.0f, 4.0f);  glVertex2f(1.0f, 2.0f);  glEnd();  //  //door  //  glBegin(GL\_POLYGON);  glColor3f(0.9f, 0.6f, 0.0f);  glVertex2f(1.0f, -4.0f);  glVertex2f(3.0f, -4.0f);  glVertex2f(3.0f, -4.0f);  glVertex2f(3.0f, 0.0f);  glVertex2f(3.0f, 0.0f);  glVertex2f(1.0f, 0.0f);  glVertex2f(1.0f, 0.0f);  glVertex2f(1.0f, -4.0f);  glEnd();  //  //roof1  //  glBegin(GL\_POLYGON);  glColor3f(0.212f, 0.275f, 0.243f);  glVertex2f(-6.0f, 6.0f);  glVertex2f(6.0f, 6.0f);  glVertex2f(6.0f, 6.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(-4.0f, 8.0f);  glVertex2f(-4.0f, 8.0f);  glVertex2f(-6.0f, 6.0f);  glEnd();  //  //floor3-4  //  glBegin(GL\_POLYGON);  glColor3f(0.659f, 0.831f, 0.49f);  glVertex2f(-4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(-4.0f, 8.0f);  glEnd();  //  //floor 3-4-outline  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(-4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 8.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(-4.0f, 8.0f);  glEnd();  //  //win1  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(-3.0f, 14.0f);  glVertex2f(-1.0f, 14.0f);  glVertex2f(-1.0f, 14.0f);  glVertex2f(-1.0f, 16.0f);  glVertex2f(-1.0f, 16.0f);  glVertex2f(-3.0f, 16.0f);  glVertex2f(-3.0f, 16.0f);  glVertex2f(-3.0f, 14.0f);  glEnd();  //  //win2  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(1.0f, 14.0f);  glVertex2f(3.0f, 14.0f);  glVertex2f(3.0f, 14.0f);  glVertex2f(3.0f, 16.0f);  glVertex2f(3.0f, 16.0f);  glVertex2f(1.0f, 16.0f);  glVertex2f(1.0f, 16.0f);  glVertex2f(1.0f, 14.0f);  glEnd();  //  //win3  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(1.0f, 10.0f);  glVertex2f(3.0f, 10.0f);  glVertex2f(3.0f, 10.0f);  glVertex2f(3.0f, 12.0f);  glVertex2f(3.0f, 12.0f);  glVertex2f(1.0f, 12.0f);  glVertex2f(1.0f, 12.0f);  glVertex2f(1.0f, 10.0f);  glEnd();  //  //win4  //  glBegin(GL\_POLYGON);  glColor3f(0.404f, 0.663f, 0.765f);  glVertex2f(-3.0f, 10.0f);  glVertex2f(-1.0f, 10.0f);  glVertex2f(-1.0f, 10.0f);  glVertex2f(-1.0f, 12.0f);  glVertex2f(-1.0f, 12.0f);  glVertex2f(-3.0f, 12.0f);  glVertex2f(-3.0f, 12.0f);  glVertex2f(-3.0f, 10.0f);  glEnd();  //  //main roof  //  glBegin(GL\_POLYGON);  glColor3f(0.212f, 0.275f, 0.243f);  glVertex2f(-4.0f, 18.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(4.0f, 18.0f);  glVertex2f(0.0f, 22.0f);  glVertex2f(0.0f, 22.0f);  glVertex2f(-4.0f, 18.0f);  glEnd();  }  void lamp() {  //  //main stand  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(12.0f, -4.0f);  glVertex2f(13.0f, -4.0f);  glVertex2f(13.0f, -4.0f);  glVertex2f(13.0f, 10.0f);  glVertex2f(13.0f, 10.0f);  glVertex2f(12.0f, 10.0f);  glVertex2f(12.0f, 10.0f);  glVertex2f(12.0f, -4.0f);  glEnd();  //  //side stand  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(11.6f, 9.0f);  glVertex2f(17.4f, 9.0f);  glVertex2f(17.4f, 9.0f);  glVertex2f(17.4f, 9.4f);  glVertex2f(17.0f, 9.4f);  glVertex2f(11.6f, 9.4f);  glVertex2f(11.6f, 9.4f);  glVertex2f(11.6f, 9.0f);  glEnd();  //  //light stand  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(15.4f, 7.8f);  glVertex2f(17.0f, 7.8f);  glVertex2f(17.0f, 7.8f);  glVertex2f(16.2f, 9.0f);  glVertex2f(16.2f, 9.0f);  glVertex2f(15.4f, 7.8f);  glEnd();  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(255, 255, 0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=0.35;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+16.25, y+7.41);  }  glEnd();  }  void bench() {  //  //side stand-1  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(14.0f, -4.0f);  glVertex2f(14.2f, -4.0f);  glVertex2f(14.2f, -4.0f);  glVertex2f(14.2f, 1.0f);  glVertex2f(14.2f, 1.0f);  glVertex2f(14.0f, 1.0f);  glVertex2f(14.0f, 1.0f);  glVertex2f(14.0f, -4.0f);  glEnd();  //  //side stand-2  //  glBegin(GL\_POLYGON);  glColor3f(0.74f, 0.74f, 0.66f);  glVertex2f(21.8f, -4.0f);  glVertex2f(22.0f, -4.0f);  glVertex2f(22.0f, -4.0f);  glVertex2f(22.0f, 1.0f);  glVertex2f(22.0f, 1.0f);  glVertex2f(21.8f, 1.0f);  glVertex2f(21.8f, 1.0f);  glVertex2f(21.8f, -4.0f);  glEnd();  //  //bench-1  //  glBegin(GL\_POLYGON);  glColor3f(0.9f, 0.6f, 0.0f);  glVertex2f(13.5f, -3.0f);  glVertex2f(22.5f, -3.0f);  glVertex2f(22.5f, -3.0f);  glVertex2f(22.5f, -2.5f);  glVertex2f(22.5f, -2.5f);  glVertex2f(13.5f, -2.5f);  glVertex2f(13.5f, -2.5f);  glVertex2f(13.5f, -3.0f);  glEnd();  //  //bench-1-outline  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(13.5f, -3.0f);  glVertex2f(22.5f, -3.0f);  glVertex2f(22.5f, -3.0f);  glVertex2f(22.5f, -2.5f);  glVertex2f(22.5f, -2.5f);  glVertex2f(13.5f, -2.5f);  glVertex2f(13.5f, -2.5f);  glVertex2f(13.5f, -3.0f);  glEnd();  //  //bench-2  //  glBegin(GL\_POLYGON);  glColor3f(0.9f, 0.6f, 0.0f);  glVertex2f(13.5f, -1.5f);  glVertex2f(22.5f, -1.5f);  glVertex2f(22.5f, -1.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(13.5f, -0.5f);  glVertex2f(13.5f, -1.5f);  glEnd();  //  //bench-2--outine  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(13.5f, -1.5f);  glVertex2f(22.5f, -1.5f);  glVertex2f(22.5f, -1.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(13.5f, -0.5f);  glVertex2f(13.5f, -1.5f);  glEnd();  //  //bench-3  //  glBegin(GL\_POLYGON);  glColor3f(0.9f, 0.6f, 0.0f);  glVertex2f(13.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, 0.5f);  glVertex2f(22.5f, 0.5f);  glVertex2f(13.5f, 0.5f);  glVertex2f(13.5f, 0.5f);  glVertex2f(13.5f, -0.5f);  glEnd();  //  //bench-3-outline  //  glBegin(GL\_LINES);  glColor3f(0.0f, 0.0f, 0.0f);  glVertex2f(13.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, -0.5f);  glVertex2f(22.5f, 0.5f);  glVertex2f(22.5f, 0.5f);  glVertex2f(13.5f, 0.5f);  glVertex2f(13.5f, 0.5f);  glVertex2f(13.5f, -0.5f);  glEnd();  }  void display() {  glClear(GL\_COLOR\_BUFFER\_BIT);  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  tree();  building();  lamp();  bench();  glFlush();  }  /\* 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(720, 720); // Set the window's initial width & height  glutInitWindowPosition(20, 20); // Position the window's initial top-left corner  glutDisplayFunc(display); // Register callback handler for window re-paint event  gluOrtho2D(-30,30,-30,30);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-** |