**Lab Taks-2**

Submission Guidelines-

* Rename the file with your serial number only.
* Must submit within time that will be discussed in class VUES to the section named Lab Tak-2
* Must include resources for all the section in the table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Question- 1**  Draw a Rainbow Flag   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |
| **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);**  **glBegin(GL\_QUADS);**  **glColor3ub(255, 0, 0); // Red**  **glVertex2f(0, 0);**  **glVertex2f(14, 0);**  **glVertex2f(14, 1);**  **glVertex2f(0, 1);**  **glColor3ub(255, 165, 0); // Orange**  **glVertex2f(0, 1);**  **glVertex2f(14, 1);**  **glVertex2f(14, 2);**  **glVertex2f(0, 2);**  **glColor3ub(255, 255, 0); // Yellow**  **glVertex2f(0, 2);**  **glVertex2f(14, 2);**  **glVertex2f(14, 3);**  **glVertex2f(0, 3);**  **glColor3ub(0, 128, 0); // Green**  **glVertex2f(0, 3);**  **glVertex2f(14, 3);**  **glVertex2f(14, 4);**  **glVertex2f(0, 4);**  **glColor3ub(0, 0, 255); // Blue**  **glVertex2f(0, 4);**  **glVertex2f(14, 4);**  **glVertex2f(14, 5);**  **glVertex2f(0, 5);**  **glColor3ub(75, 0, 130); // Indigo**  **glVertex2f(0, 5);**  **glVertex2f(14, 5);**  **glVertex2f(14, 6);**  **glVertex2f(0, 6);**  **glColor3ub(238, 130, 238); // Violet**  **glVertex2f(0, 6);**  **glVertex2f(14, 6);**  **glVertex2f(14, 7);**  **glVertex2f(0, 7);**  **glEnd();**  **glFlush();**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutCreateWindow("Rainbow Flag");**  **glutInitWindowSize(320, 320);**  **glutInitWindowPosition(50, 50);**  **gluOrtho2D(-20, 20, -20, 20);**  **glutDisplayFunc(display);**  **initGL();**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Flag of 5 different countries as per your choice. |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // For MS Windows**  **#include <GL/glut.h> // GLUT, includes glu.h and gl.h**  **void display() {**  **glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer**  **glLineWidth(10);**  **// ITALY**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 146, 70); // Green**  **glVertex2f(0.5, 0.5);**  **glVertex2f(1.5, 0.5);**  **glVertex2f(1.5, 2);**  **glVertex2f(0.5, 2);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // White**  **glVertex2f(1.5, 0.5);**  **glVertex2f(2.5, 0.5);**  **glVertex2f(2.5, 2);**  **glVertex2f(1.5, 2);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(206, 43, 55); // Red**  **glVertex2f(2.5, 0.5);**  **glVertex2f(3.5, 0.5);**  **glVertex2f(3.5, 2);**  **glVertex2f(2.5, 2);**  **glEnd();**  **// POLAND**  **glBegin(GL\_POLYGON);**  **glColor3ub(220, 20, 60); // Red**  **glVertex2f(2, 3);**  **glVertex2f(6, 3);**  **glVertex2f(6, 4);**  **glVertex2f(2, 4);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // White**  **glVertex2f(2, 4);**  **glVertex2f(6, 4);**  **glVertex2f(6, 5);**  **glVertex2f(2, 5);**  **glEnd();**  **// GERMANY**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 206, 0); // Yellow**  **glVertex2f(-7, 1);**  **glVertex2f(-1, 1);**  **glVertex2f(-1, 2);**  **glVertex2f(-7, 2);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(221, 0, 0); // Red**  **glVertex2f(-7, 2);**  **glVertex2f(-1, 2);**  **glVertex2f(-1, 3);**  **glVertex2f(-7, 3);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Black**  **glVertex2f(-7, 3);**  **glVertex2f(-1, 3);**  **glVertex2f(-1, 4);**  **glVertex2f(-7, 4);**  **glEnd();**  **// RUSSIA**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // White**  **glVertex2f(-7, -1);**  **glVertex2f(-1, -1);**  **glVertex2f(-1, -2);**  **glVertex2f(-7, -2);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 57, 166); // Blue**  **glVertex2f(-7, -2);**  **glVertex2f(-1, -2);**  **glVertex2f(-1, -3);**  **glVertex2f(-7, -3);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(213, 43, 30); // Red**  **glVertex2f(-7, -3);**  **glVertex2f(-1, -3);**  **glVertex2f(-1, -4);**  **glVertex2f(-7, -4);**  **glEnd();**  **// AUSTRIA**  **glBegin(GL\_POLYGON);**  **glColor3ub(237, 41, 57); // Red**  **glVertex2f(1, -1);**  **glVertex2f(7, -1);**  **glVertex2f(7, -2);**  **glVertex2f(1, -2);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // White**  **glVertex2f(1, -2);**  **glVertex2f(7, -2);**  **glVertex2f(7, -3);**  **glVertex2f(1, -3);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(237, 41, 57); // Red**  **glVertex2f(1, -3);**  **glVertex2f(7, -3);**  **glVertex2f(7, -4);**  **glVertex2f(1, -4);**  **glEnd();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv); // Initialize GLUT**  **glutCreateWindow("OpenGL Flags"); // Create a window**  **glutInitWindowSize(500, 500); // Set window size**  **gluOrtho2D(-10, 10, -10, 10); // Set 2D coordinate system**  **glutDisplayFunc(display); // Register display function**  **glutMainLoop(); // Enter event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 3**  Draw 4X4 Chess Board |
| **Graph Plot (Picture)-** |
| **Code-**  **#include <windows.h> // For MS Windows**  **#include <GL/glut.h> // GLUT, includes glu.h and gl.h**  **void initGL() {**  **glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // White background**  **}**  **void display() {**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer**  **// Manually draw each square**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glBegin(GL\_QUADS);**  **glVertex2f(0, 0);**  **glVertex2f(1, 0);**  **glVertex2f(1, 1);**  **glVertex2f(0, 1);**  **glEnd();**  **glColor3f(1.0f, 1.0f, 1.0f); // White**  **glBegin(GL\_QUADS);**  **glVertex2f(1, 0);**  **glVertex2f(2, 0);**  **glVertex2f(2, 1);**  **glVertex2f(1, 1);**  **glEnd();**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glBegin(GL\_QUADS);**  **glVertex2f(2, 0);**  **glVertex2f(3, 0);**  **glVertex2f(3, 1);**  **glVertex2f(2, 1);**  **glEnd();**  **glColor3f(1.0f, 1.0f, 1.0f); // White**  **glBegin(GL\_QUADS);**  **glVertex2f(3, 0);**  **glVertex2f(4, 0);**  **glVertex2f(4, 1);**  **glVertex2f(3, 1);**  **glEnd();**  **glColor3f(1.0f, 1.0f, 1.0f); // White**  **glBegin(GL\_QUADS);**  **glVertex2f(0, 1);**  **glVertex2f(1, 1);**  **glVertex2f(1, 2);**  **glVertex2f(0, 2);**  **glEnd();**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glBegin(GL\_QUADS);**  **glVertex2f(1, 1);**  **glVertex2f(2, 1);**  **glVertex2f(2, 2);**  **glVertex2f(1, 2);**  **glEnd();**  **glColor3f(1.0f, 1.0f, 1.0f); // White**  **glBegin(GL\_QUADS);**  **glVertex2f(2, 1);**  **glVertex2f(3, 1);**  **glVertex2f(3, 2);**  **glVertex2f(2, 2);**  **glEnd();**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glBegin(GL\_QUADS);**  **glVertex2f(3, 1);**  **glVertex2f(4, 1);**  **glVertex2f(4, 2);**  **glVertex2f(3, 2);**  **glEnd();**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glBegin(GL\_QUADS);**  **glVertex2f(0, 2);**  **glVertex2f(1, 2);**  **glVertex2f(1, 3);**  **glVertex2f(0, 3);**  **glEnd();**  **glColor3f(1.0f, 1.0f, 1.0f); // White**  **glBegin(GL\_QUADS);**  **glVertex2f(1, 2);**  **glVertex2f(2, 2);**  **glVertex2f(2, 3);**  **glVertex2f(1, 3);**  **glEnd();**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glBegin(GL\_QUADS);**  **glVertex2f(2, 2);**  **glVertex2f(3, 2);**  **glVertex2f(3, 3);**  **glVertex2f(2, 3);**  **glEnd();**  **glColor3f(1.0f, 1.0f, 1.0f); // White**  **glBegin(GL\_QUADS);**  **glVertex2f(3, 2);**  **glVertex2f(4, 2);**  **glVertex2f(4, 3);**  **glVertex2f(3, 3);**  **glEnd();**  **glColor3f(1.0f, 1.0f, 1.0f); // White**  **glBegin(GL\_QUADS);**  **glVertex2f(0, 3);**  **glVertex2f(1, 3);**  **glVertex2f(1, 4);**  **glVertex2f(0, 4);**  **glEnd();**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glBegin(GL\_QUADS);**  **glVertex2f(1, 3);**  **glVertex2f(2, 3);**  **glVertex2f(2, 4);**  **glVertex2f(1, 4);**  **glEnd();**  **glColor3f(1.0f, 1.0f, 1.0f); // White**  **glBegin(GL\_QUADS);**  **glVertex2f(2, 3);**  **glVertex2f(3, 3);**  **glVertex2f(3, 4);**  **glVertex2f(2, 4);**  **glEnd();**  **glColor3f(0.0f, 0.0f, 0.0f); // Black**  **glBegin(GL\_QUADS);**  **glVertex2f(3, 3);**  **glVertex2f(4, 3);**  **glVertex2f(4, 4);**  **glVertex2f(3, 4);**  **glEnd();**  **glFlush(); // Render now**  **}**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutCreateWindow("4x4 Chess Board");**  **glutInitWindowSize(400, 400);**  **glutInitWindowPosition(80, 80);**  **gluOrtho2D(0, 4, 0, 4); // Define the coordinate system**  **glutDisplayFunc(display);**  **initGL();**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 4**  Improvement icon in comic style. Gear project cartoon vector illustration  on white isolated background. Productivity splash effect business concept.  16134530 Vector Art at VecteezyImplement the given picture |
| **Graph Plot (Picture)-** |
|  |
| **Code-**  **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **#include <math.h>**  **// Function to display graphics**  **void display() {**  **glClearColor(1.0, 1.0, 1.0, 1.0); // Set background color to white**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer**  **glLineWidth(3);**  **// Define positions for four circles**  **float positions[4][2] = {**  **{-0.3699, 0.3504}, {0.3436, 0.3677},**  **{0.3436, -0.3401}, {-0.3526, -0.3516}**  **};**  **float r = 0.5;**  **// Draw four circles**  **for (int j = 0; j < 4; j++) {**  **glBegin(GL\_POLYGON);**  **glColor3f(223.0 / 255, 243.0 / 255, 252.0 / 255);**  **for (int i = 0; i < 200; i++) {**  **float A = (i \* 2 \* M\_PI) / 200;**  **glVertex2f(r \* cos(A) + positions[j][0], r \* sin(A) + positions[j][1]);**  **}**  **glEnd();**  **}**  **//arrow**  **glBegin(GL\_QUADS);**  **glColor3ub(246,146,148);**  **glVertex2f(-0.1,0.2);**  **glVertex2f(-0.1,0.4);**  **glVertex2f(0.1,0.4);**  **glVertex2f(0.1,0.2);**  **glEnd();**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(246,146,148);**  **glVertex2f(-0.2,0.4);**  **glVertex2f(0,0.6);**  **glVertex2f(0.2,0.4);**  **glEnd();**  **// Draw center circles**  **float center[2] = {0.0, 0.1};**  **float R = 0.1;**  **for (int j = 0; j < 2; j++) {**  **glBegin(GL\_POLYGON);**  **glColor3f(166.0 / 255, 223.0 / 255, 250.0 / 255);**  **for (int i = 0; i < 200; i++) {**  **float A = (i \* 2 \* M\_PI) / 200;**  **glVertex2f(R \* cos(A) + center[0], R \* sin(A) + center[1]);**  **}**  **R += 0.05;**  **glEnd();**  **}**  **// Draw triangle**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(250, 208, 132);**  **glVertex2f(-0.8, 0.6);**  **glVertex2f(-0.7, 0.55);**  **glVertex2f(-0.8, 0.5);**  **glEnd();**  **// Draw plus sign (+)**  **glBegin(GL\_LINES);**  **glColor3ub(247, 146, 146);**  **glVertex2f(-0.25, 0.75);**  **glVertex2f(-0.15, 0.75);**  **glVertex2f(-0.2, 0.8);**  **glVertex2f(-0.2, 0.7);**  **glEnd();**  **// Draw another triangle**  **glBegin(GL\_TRIANGLES);**  **glColor3ub(241, 146, 144);**  **glVertex2f(0.1, -0.75);**  **glVertex2f(0.2, -0.7);**  **glVertex2f(0.2, -0.8);**  **glEnd();**  **// Draw cross sign (X)**  **glBegin(GL\_LINES);**  **glColor3ub(255, 152, 153);**  **glVertex2f(-0.6, -0.6);**  **glVertex2f(-0.5, -0.7);**  **glVertex2f(-0.6, -0.7);**  **glVertex2f(-0.5, -0.6);**  **glEnd();**  **// Define square positions**  **float squares[3][4][2] = {**  **{{-0.4, -0.5}, {-0.4, -0.3}, {-0.2, -0.3}, {-0.2, -0.5}},**  **{{-0.1, -0.5}, {-0.1, -0.3}, {0.1, -0.3}, {0.1, -0.5}},**  **{{0.2, -0.5}, {0.2, -0.3}, {0.4, -0.3}, {0.4, -0.5}}**  **};**  **// Draw squares**  **glColor3ub(254, 211, 133);**  **for (int i = 0; i < 3; i++) {**  **glBegin(GL\_QUADS);**  **for (int j = 0; j < 4; j++) {**  **glVertex2f(squares[i][j][0], squares[i][j][1]);**  **}**  **glEnd();**  **}**  **glBegin(GL\_QUADS);**  **glColor3ub(254, 211, 133);**  **glVertex2f(-0.32,-0.3);**  **glVertex2f(-0.32,0.12);**  **glVertex2f(-0.28,0.12);**  **glVertex2f(-0.28,-0.3);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(254, 211, 133);**  **glVertex2f(-0.28,0.08);**  **glVertex2f(-0.28,0.12);**  **glVertex2f(-0.1485776339369, 0.1206079279332);**  **glVertex2f(-0.1486862465207, 0.0801909087643);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(254, 211, 133);**  **glVertex2f(-0.02,-0.3);**  **glVertex2f(-0.0200669441337,-0.0482869536007);**  **glVertex2f(0.0199211307485,-0.0482869536007);**  **glVertex2f(0.02,-0.3);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(254, 211, 133);**  **glVertex2f(0.28,-0.3);**  **glVertex2f(0.28,0.1);**  **glVertex2f(0.32,0.1);**  **glVertex2f(0.32,-0.3);**  **glEnd();**  **glBegin(GL\_QUADS);**  **glColor3ub(254, 211, 133);**  **glVertex2f(0.1452852045516,0.0598556045804);**  **glVertex2f(0.1493268540589,0.0998230274851);**  **glVertex2f(0.28,0.1);**  **glVertex2f(0.28,0.06);**  **glEnd();**  **glFlush();**  **}**  **// Main function**  **int main(int argc, char\*\* argv) {**  **glutInit(&argc, argv);**  **glutCreateWindow("OpenGL Shapes");**  **glutInitWindowSize(501, 495);**  **gluOrtho2D(-1, 1, -1, 1);**  **glutDisplayFunc(display);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-** |