**Lab Taks-2**

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-2
* Must include resources for all the section in the table

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Question- 1**  Draw a Rainbow Flag   |  | | --- | |  | |  | |  | |  | |  | |  | |  | |
| **Graph Plot (Picture)-** |
| **Code-** **#include <windows.h> // for MS Windows**  **#include <GL/glut.h>**  **void userdraw(void);**  **void drawDot(float x, float y)**  **{**  **glBegin(GL\_POINTS) ;**  **glVertex2f(x,y) ;**  **glEnd() ;**  **}**  **void drawLine(float x1,float y1, float x2, float y2){**  **glBegin(GL\_LINES);**  **glVertex2f(x1,y1);**  **glVertex2f(x2,y2);**  **glEnd();**  **}**  **void setColor(float red, float green, float blue){**  **glColor3f(red, green, blue);**  **}**  **void userdraw(void){**  **glLineWidth(10.0f);**  **glBegin(GL\_LINES);**  **glColor4f(1.0f, 0.0f, 1.0f, 0.0f);**  **drawLine(10.,190.,230.,190);**  **glColor3f(0.0f, 0.0f, 1.0f); //orange**  **drawLine(10.,180.,230.,180);**  **glColor4f(0.0f, 1.0f, 1.0f, 1.0f);**  **drawLine(10.,170.,230.,170);**  **glColor3f(0.0f, 1.0f, 0.0f);**  **drawLine(10.,160.,230.,160);**  **glColor4f(1.0f, 1.0f, 0.0f, 0.0f);//yellow**  **drawLine(10.,150.,230.,150);**  **glColor4f(1.0f, 0.5f, 0.0f, 0.0f);//orange**  **drawLine(10.,140.,230.,140);**  **glColor4f(1.0f, 0.0f, 0.0f, 0.0f);//red**  **drawLine(10.,130.,230.,130);**  **}**  **void display (void) {**  **//clear screen**  **glClear(GL\_COLOR\_BUFFER\_BIT);**  **userdraw();**  **glutSwapBuffers();**  **}**  **int main (int argc, char \*\*argv){**  **glutInit( &argc,argv);**  **glutInitDisplayMode( GLUT\_DOUBLE | GLUT\_RGB );**  **glutInitWindowSize(640,480);**  **glutInitWindowPosition(100,150);**  **glutCreateWindow("Rainbow");**  **glClearColor(1.0,1.0,1.0,0.0);**  **gluOrtho2D(0.,640.,-240.,240.);**  **glutIdleFunc(display);**  **glutDisplayFunc(display);**  **glutMainLoop();**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-RAINBOW.png** |

|  |
| --- |
| **Question- 2**  Draw 8X8 Chess Board |
| **Graph Plot (Picture)-** |
| **Code-** **#include <windows.h> // for MS Windows**  **#include <GL/glut.h> // GLUT, include glu.h and gl.h**  **/\* Handler for window-repaint event. Call back when the window first appears and**  **whenever the window needs to be re-painted. \*/**  **void display() {**  **glClearColor(255.0f, 255.0f, 255.0f, 1.0f); // Set background color to white and opaque**  **glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)**  **glLineWidth(2);**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // Each set of 4 vertices form a quad// x, y**  **glVertex2f(0.0f,1.0f);**  **glVertex2f(1.0f,1.0f);**  **glVertex2f(1.0f,2.0f);**  **glVertex2f(0.0f,2.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(1.0f,1.0f);**  **glVertex2f(2.0f,1.0f);**  **glVertex2f(2.0f,2.0f);**  **glVertex2f(1.0f,2.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(2.0f,1.0f);**  **glVertex2f(3.0f,1.0f);**  **glVertex2f(3.0f,2.0f);**  **glVertex2f(2.0f,2.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(3.0f,1.0f);**  **glVertex2f(4.0f,1.0f);**  **glVertex2f(4.0f,2.0f);**  **glVertex2f(3.0f,2.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(0.0f,2.0f);**  **glVertex2f(1.0f,2.0f);**  **glVertex2f(1.0f,3.0f);**  **glVertex2f(0.0f,3.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(1.0f,2.0f);**  **glVertex2f(2.0f,2.0f);**  **glVertex2f(2.0f,3.0f);**  **glVertex2f(1.0f,3.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad// x, y**  **glVertex2f(2.0f,2.0f);**  **glVertex2f(3.0f,2.0f);**  **glVertex2f(3.0f,3.0f);**  **glVertex2f(2.0f,3.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(3.0f,2.0f);**  **glVertex2f(4.0f,2.0f);**  **glVertex2f(4.0f,3.0f);**  **glVertex2f(3.0f,3.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(0.0f,3.0f);**  **glVertex2f(1.0f,3.0f);**  **glVertex2f(1.0f,4.0f);**  **glVertex2f(0.0f,4.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(1.0f,3.0f);**  **glVertex2f(2.0f,3.0f);**  **glVertex2f(2.0f,4.0f);**  **glVertex2f(1.0f,4.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(2.0f,3.0f);**  **glVertex2f(3.0f,3.0f);**  **glVertex2f(3.0f,4.0f);**  **glVertex2f(2.0f,4.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(3.0f,3.0f);**  **glVertex2f(4.0f,3.0f);**  **glVertex2f(4.0f,4.0f);**  **glVertex2f(3.0f,4.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad// x, y**  **glVertex2f(0.0f,4.0f);**  **glVertex2f(1.0f,4.0f);**  **glVertex2f(1.0f,5.0f);**  **glVertex2f(0.0f,5.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // Each set of 4 vertices form a quad// x, y**  **glVertex2f(1.0f,4.0f);**  **glVertex2f(2.0f,4.0f);**  **glVertex2f(2.0f,5.0f);**  **glVertex2f(1.0f,5.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad // x, y**  **glVertex2f(2.0f,4.0f);**  **glVertex2f(3.0f,4.0f);**  **glVertex2f(3.0f,5.0f);**  **glVertex2f(2.0f,5.0f);**  **glEnd();**  **glBegin(GL\_POLYGON);**  **glColor3ub(255, 255, 255); // Each set of 4 vertices form a quad// x, y**  **glVertex2f(3.0f,4.0f);**  **glVertex2f(4.0f,4.0f);**  **glVertex2f(4.0f,5.0f);**  **glVertex2f(3.0f,5.0f);**  **glEnd();**  **// Draw a border**  **glBegin(GL\_LINES);**  **glColor3ub(0, 0, 0); // Each set of 4 vertices form a quad// x, y**  **glVertex2f(0.0f,1.0f);**  **glVertex2f(4.0f,1.0f);**  **glVertex2f(4.0f,1.0f);**  **glVertex2f(4.0f,5.0f);**  **glVertex2f(4.0f,5.0f);**  **glVertex2f(0.0f,5.0f);**  **glVertex2f(0.0f,5.0f);**  **glVertex2f(0.0f,1.0f);**  **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("OpenGL Setup Test"); // Create a window with the given title**  **glutInitWindowSize(320, 320);**  **gluOrtho2D(-2,6,-2,6); // Set the window's initial width & height**  **glutDisplayFunc(display); // Register display callback handler for window re-paint**  **glutMainLoop(); // Enter the event-processing loop**  **return 0;**  **}** |
| **Output Screenshot (Full Screen)-**  **CHESS BOARD.png** |

|  |
| --- |
| **Question- 3**  Create the batman logo given below- |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **Code-**  **Code-#include <windows.h>**  **#include <GL/glut.h>**  **void Display(void) {**  **glClear (GL\_COLOR\_BUFFER\_BIT);**  **glColor3ub (255, 255, 255);**  **glBegin(GL\_QUADS);**  **glVertex2i(0, 0);**  **glVertex2i(640, 0);**  **glVertex2i(640, 480);**  **glVertex2i(0, 480);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(427, 440);**  **glVertex2i(427, 460);**  **glVertex2i(212, 460);**  **glVertex2i(212, 440);**  **glEnd(); glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(212, 414);**  **glVertex2i(212, 440);**  **glVertex2i(148, 440);**  **glVertex2i(148, 414);**  **glEnd(); glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(148, 390);**  **glVertex2i(148, 414);**  **glVertex2i(118, 414);**  **glVertex2i(118, 390);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(118, 362);**  **glVertex2i(118,390);**  **glVertex2i(90, 390);**  **glVertex2i(90, 362);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(90, 337);**  **glVertex2i(90,362);**  **glVertex2i(55, 362);**  **glVertex2i(55, 337);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(55, 156);**  **glVertex2i(55,337);**  **glVertex2i(25, 337);**  **glVertex2i(25, 156);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(82, 130);**  **glVertex2i(82,156);**  **glVertex2i(55, 156);**  **glVertex2i(55, 130);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(114, 105);**  **glVertex2i(114,130);**  **glVertex2i(82, 130);**  **glVertex2i(82, 105);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(145, 80);**  **glVertex2i(145,105);**  **glVertex2i(114, 105);**  **glVertex2i(114, 80);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(210, 55);**  **glVertex2i(210,80);**  **glVertex2i(145, 80);**  **glVertex2i(145, 55);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(427, 30);**  **glVertex2i(427,55);**  **glVertex2i(212, 55);**  **glVertex2i(212, 30);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(490, 55);**  **glVertex2i(490,80);**  **glVertex2i(427, 80);**  **glVertex2i(427, 55);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(520, 80);**  **glVertex2i(520,102);**  **glVertex2i(490, 102);**  **glVertex2i(490, 80);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(550, 102);**  **glVertex2i(550,128);**  **glVertex2i(520, 128);**  **glVertex2i(520, 102);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(580, 128);**  **glVertex2i(580,156);**  **glVertex2i(550, 156);**  **glVertex2i(550, 128);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(610, 156);**  **glVertex2i(610,337);**  **glVertex2i(580, 337);**  **glVertex2i(580, 156);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(580, 337);**  **glVertex2i(580,360);**  **glVertex2i(555, 360);**  **glVertex2i(555, 337);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(555, 360);**  **glVertex2i(555,385);**  **glVertex2i(523, 385);**  **glVertex2i(523, 360);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(523, 385);**  **glVertex2i(523,410);**  **glVertex2i(490, 410);**  **glVertex2i(490, 385);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(490, 410);**  **glVertex2i(490,440);**  **glVertex2i(427, 440);**  **glVertex2i(427, 410);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(580, 156);**  **glVertex2i(580,337);**  **glVertex2i(555, 337);**  **glVertex2i(550, 156);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(550, 128);**  **glVertex2i(556,360);**  **glVertex2i(523, 360);**  **glVertex2i(520, 128);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(520, 102);**  **glVertex2i(524,385);**  **glVertex2i(490, 385);**  **glVertex2i(490, 102);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(490, 80);**  **glVertex2i(490,410);**  **glVertex2i(427, 410);**  **glVertex2i(427, 80);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(427, 440);**  **glVertex2i(427,55);**  **glVertex2i(212, 55);**  **glVertex2i(212, 440);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(212, 80);**  **glVertex2i(212,414);**  **glVertex2i(145, 414);**  **glVertex2i(145, 80);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(145, 105);**  **glVertex2i(148,390);**  **glVertex2i(118, 390);**  **glVertex2i(114, 105);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(115, 130);**  **glVertex2i(118,362);**  **glVertex2i(90, 362);**  **glVertex2i(82, 130);**  **glEnd();**  **glColor3ub (255, 255, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(83, 156);**  **glVertex2i(90,337);**  **glVertex2i(55, 337);**  **glVertex2i(55, 156);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(550, 182);**  **glVertex2i(550, 306);**  **glVertex2i(523, 306);**  **glVertex2i(523, 182);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(523, 157);**  **glVertex2i(523, 334);**  **glVertex2i(490, 334);**  **glVertex2i(490, 157);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(490, 130);**  **glVertex2i(490, 360);**  **glVertex2i(460, 360);**  **glVertex2i(460, 130);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(460, 180);**  **glVertex2i(460, 310);**  **glVertex2i(430, 310);**  **glVertex2i(430, 180);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(430, 155);**  **glVertex2i(430, 280);**  **glVertex2i(400, 280);**  **glVertex2i(400, 155);**  **glEnd();**  **glColor3ub (0, 0, 0);**  **glBegin(GL\_QUADS);**  **glVertex2i(400, 205);**  **glVertex2i(400, 310);**  **glVertex2i(365, 310);**  **glVertex2i(365, 205);**  **glEnd();**  **glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(365, 155); glVertex2i(365, 360); glVertex2i(275, 360); glVertex2i(275, 155); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(275, 205);**  **glVertex2i(275, 310); glVertex2i(245, 310); glVertex2i(245, 205); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(245, 155);**  **glVertex2i(245, 280); glVertex2i(215, 280); glVertex2i(215, 155); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(215, 180); glVertex2i(215, 310); glVertex2i(185, 310); glVertex2i(185, 180); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(185, 130); glVertex2i(185, 360);**  **glVertex2i(155, 360); glVertex2i(155, 130); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(155, 157); glVertex2i(155, 334);**  **glVertex2i(125, 334); glVertex2i(125, 157); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(125, 182); glVertex2i(125, 306);**  **glVertex2i(95, 306); glVertex2i(95, 182); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(350, 105); glVertex2i(350, 360);**  **glVertex2i(290, 360); glVertex2i(290, 105); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(335, 80); glVertex2i(335, 105);**  **glVertex2i(305, 105); glVertex2i(305, 80); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(365, 360); glVertex2i(365, 385);**  **glVertex2i(335, 385); glVertex2i(335, 360); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(305, 360); glVertex2i(305, 385);**  **glVertex2i(275, 385); glVertex2i(275, 360); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(460, 345); glVertex2i(460, 370);**  **glVertex2i(430, 370); glVertex2i(430, 345); glEnd(); glColor3ub (0, 0, 0); glBegin(GL\_QUADS); glVertex2i(210, 345); glVertex2i(210, 370);**  **glVertex2i(180, 370); glVertex2i(180, 345); glEnd(); glFlush (); }**  **void myInit (void) { glClearColor(0.0, 0.0, 0.0, 0.0); glMatrixMode(GL\_PROJECTION); glLoadIdentity(); gluOrtho2D(0.0, 640.0, 0.0, 480.0); }**  **int main(int argc, char\*\* argv) { glutInit(&argc, argv);**  **glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);**  **glutInitWindowSize (640, 480);**  **glutInitWindowPosition (100, 150);**  **glutCreateWindow ("Batman"); glutDisplayFunc(Display); myInit ();**  **glutMainLoop();**  **return 0; }** |
| **Output Screenshot (Full Screen)-**  BATMAN.png |