**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)-**  **labtask2(a)** |
| **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(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  glLineWidth(4.0);  // Draw a Red 1x1 Square centered at origin  glBegin(GL\_QUADS);  glColor3ub(238,130,238);  glVertex2d(-80.0,80.0);  glVertex2d(-80.0,70.0);  glVertex2d(80.0,70.0);  glVertex2d(80.0,80.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(0, 0, 255);  glVertex2d(-80.0,70.0);  glVertex2d(-80.0,60.0);  glVertex2d(80.0,60.0);  glVertex2d(80.0,70.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(135,206,235);  glVertex2d(-80.0,60.0);  glVertex2d(-80.0,50.0);  glVertex2d(80.0,50.0);  glVertex2d(80.0,60.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(0,128,0);  glVertex2d(-80.0,50.0);  glVertex2d(-80.0,40.0);  glVertex2d(80.0,40.0);  glVertex2d(80.0,50.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,255,0);  glVertex2d(-80.0,40.0);  glVertex2d(-80.0,30.0);  glVertex2d(80.0,30.0);  glVertex2d(80.0,40.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,165,0);  glVertex2d(-80.0,30.0);  glVertex2d(-80.0,20.0);  glVertex2d(80.0,20.0);  glVertex2d(80.0,30.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255, 0, 0);  glVertex2d(-80.0,20.0);  glVertex2d(-80.0,10.0);  glVertex2d(80.0,10.0);  glVertex2d(80.0,20.0);  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  glutInitWindowSize(520, 520); // Set the window's initial width & height  glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title  gluOrtho2D(-100,100,-100,100);  glutDisplayFunc(display); // Register display callback handler for window re-paint  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-lasbtask2(a)** |

|  |
| --- |
| **Question- 2**  Draw 8X8 Chess Board |
| **Graph Plot (Picture)-**  **labtask2(b)** |
| **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(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  glLineWidth(4.0);  // Draw a Red 1x1 Square centered at origin  glBegin(GL\_QUADS);  glColor3ub(255,255,255);  glVertex2d(-80.0,70.0);  glVertex2d(-80.0,60.0);  glVertex2d(-70.0,60.0);  glVertex2d(-70.0,70.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(128,128,128);  glVertex2d(-70.0,70.0);  glVertex2d(-70.0,60.0);  glVertex2d(-60.0,60.0);  glVertex2d(-60.0,70.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,255,255);  glVertex2d(-60.0,70.0);  glVertex2d(-60.0,60.0);  glVertex2d(-50.0,60.0);  glVertex2d(-50.0,70.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(128,128,128);  glVertex2d(-50.0,70.0);  glVertex2d(-50.0,60.0);  glVertex2d(-40.0,60.0);  glVertex2d(-40.0,70.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(128,128,128);  glVertex2d(-80.0,60.0);  glVertex2d(-80.0,50.0);  glVertex2d(-70.0,50.0);  glVertex2d(-70.0,60.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,255,255);  glVertex2d(-70.0,60.0);  glVertex2d(-70.0,50.0);  glVertex2d(-60.0,50.0);  glVertex2d(-60.0,60.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(128,128,128);  glVertex2d(-60.0,60.0);  glVertex2d(-60.0,50.0);  glVertex2d(-50.0,50.0);  glVertex2d(-50.0,60.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,255,255);  glVertex2d(-50.0,60.0);  glVertex2d(-50.0,50.0);  glVertex2d(-40.0,50.0);  glVertex2d(-40.0,60.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,255,255);  glVertex2d(-80.0,50.0);  glVertex2d(-80.0,40.0);  glVertex2d(-70.0,40.0);  glVertex2d(-70.0,50.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(128,128,128);  glVertex2d(-70.0,50.0);  glVertex2d(-70.0,40.0);  glVertex2d(-60.0,40.0);  glVertex2d(-60.0,50.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,255,255);  glVertex2d(-60.0,50.0);  glVertex2d(-60.0,40.0);  glVertex2d(-50.0,40.0);  glVertex2d(-50.0,50.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(128,128,128);  glVertex2d(-50.0,50.0);  glVertex2d(-50.0,40.0);  glVertex2d(-40.0,40.0);  glVertex2d(-40.0,50.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(128,128,128);  glVertex2d(-80.0,40.0);  glVertex2d(-80.0,30.0);  glVertex2d(-70.0,30.0);  glVertex2d(-70.0,40.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,255,255);  glVertex2d(-70.0,40.0);  glVertex2d(-70.0,30.0);  glVertex2d(-60.0,30.0);  glVertex2d(-60.0,40.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(128,128,128);  glVertex2d(-60.0,40.0);  glVertex2d(-60.0,30.0);  glVertex2d(-50.0,30.0);  glVertex2d(-50.0,40.0);  glEnd();  glBegin(GL\_QUADS);  glColor3ub(255,255,255);  glVertex2d(-50.0,40.0);  glVertex2d(-50.0,30.0);  glVertex2d(-40.0,30.0);  glVertex2d(-40.0,40.0);  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  glutInitWindowSize(520, 520); // Set the window's initial width & height  glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title  gluOrtho2D(-100,100,-100,100);  glutDisplayFunc(display); // Register display callback handler for window re-paint  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-**  labtask2(b) |

|  |
| --- |
| **Question- 3**  Create the batman logo given below- |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **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(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT); // Clear the color buffer (background)  glLineWidth(4.0);  // Draw a Red 1x1 Square centered at origin  glBegin(GL\_POLYGON);  glColor3ub(0,0,0);  glVertex2d(55.0,55.0);  glVertex2d(-55.0,55.0);  glVertex2d(-80.0,35.0);  glVertex2d(-80.0,-5.0);  glVertex2d(-55.0,-25.0);  glVertex2d(55.0,-25.0);  glVertex2d(80.0,35.0);  glVertex2d(80.0,-5.0);  glVertex2d(55.0,-25.0);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,255,0);  glVertex2d(50.0,50.0);  glVertex2d(-50.0,50.0);  glVertex2d(-75.0,30.0);  glVertex2d(-75.0,0.0);  glVertex2d(-50.0,-20.0);  glVertex2d(50.0,-20.0);  glVertex2d(75.0,30.0);  glVertex2d(75.0,0.0);  glVertex2d(50.0,-20.0);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(0,0,0);  glVertex2d(10,40);  glVertex2d(10,20);  glVertex2d(-10,20);  glVertex2d(-10,40);  glVertex2d(-15,40);  glVertex2d(-15,10);  glVertex2d(-30,10);  glVertex2d(-30,15);  glVertex2d(-40,15);  glVertex2d(-40,30);  glVertex2d(-45,25);  glVertex2d(-45,20);  glVertex2d(-50,20);  glVertex2d(-50,-10);  glVertex2d(-45,-10);  glVertex2d(-45,-5);  glVertex2d(-35,-5);  glVertex2d(0,-17);  glVertex2d(35,-5);  glVertex2d(45,-5);  glVertex2d(45,-10);  glVertex2d(50,-10);  glVertex2d(50,20);  glVertex2d(45,20);  glVertex2d(45,25);  glVertex2d(40,30);  glVertex2d(40,15);  glVertex2d(30,15);  glVertex2d(30,10);  glVertex2d(15,10);  glVertex2d(15,40);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,255,0);  glVertex2d(10,40);  glVertex2d(-10,40);  glVertex2d(-10,30);  glVertex2d(10,30);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,255,0);  glVertex2d(30,40);  glVertex2d(15,40);  glVertex2d(15,20);  glVertex2d(30,20);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255,255,0);  glVertex2d(-30,40);  glVertex2d(-15,40);  glVertex2d(-15,20);  glVertex2d(-30,20);  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  glutInitWindowSize(520, 520); // Set the window's initial width & height  glutCreateWindow("OpenGL Setup Test"); // Create a window with the given title  gluOrtho2D(-100,100,-100,100);  glutDisplayFunc(display); // Register display callback handler for window re-paint  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| **Output Screenshot (Full Screen)-**  **labtask2(c)** |