**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>  #include <GL/glut.h>  void display()  {  //glLoadIdentity();  glClear(GL\_COLOR\_BUFFER\_BIT);  //purple  glColor3f(101 / 255.0, 89 / 255.0, 139 / 255.0);  glBegin(GL\_POLYGON);  glVertex2f(25.0, 20.0);  glVertex2f(25.0, 15.0);  glVertex2f(-25.0, 15.0);  glVertex2f(-25.0, 20.0);  glEnd();  //blue  glColor3f(51 / 255.0, 121 / 255.0, 173 / 255.0);  glBegin(GL\_POLYGON);  glVertex2f(25.0, 15.0);  glVertex2f(25.0, 10.0);  glVertex2f(-25.0, 10.0);  glVertex2f(-25.0, 15.0);  glEnd();  //cyan  glColor3f(49 / 255.0, 152 / 255.0, 171 / 255.0);  glBegin(GL\_POLYGON);  glVertex2f(25.0, 10.0);  glVertex2f(25.0, 5.0);  glVertex2f(-25.0, 5.0);  glVertex2f(-25.0, 10.0);  glEnd();  //green  glColor3f(9 / 255.0, 114 / 255.0, 39 / 255.0);  glBegin(GL\_POLYGON);  glVertex2f(25.0, 5.0);  glVertex2f(25.0, 0.0);  glVertex2f(-25.0, 0.0);  glVertex2f(-25.0, 5.0);  glEnd();  //orange  glColor3f(194.0 / 255.0, 97.0 / 255.0, 34.0 / 255.0);  glBegin(GL\_POLYGON);  glVertex2f(25.0, 0.0);  glVertex2f(25.0, -5.0);  glVertex2f(-25.0, -5.0);  glVertex2f(-25.0, 0.0);  glEnd();  //yellow  glColor3f(1.0, 1.0, 0.0);  glBegin(GL\_POLYGON);  glVertex2f(25.0, -5.0);  glVertex2f(25.0, -10.0);  glVertex2f(-25.0, -10.0);  glVertex2f(-25.0, -5.0);  glEnd();  //red  glColor3f(1.0, 0.0, 0.0);  glBegin(GL\_POLYGON);  glVertex2f(25.0, -10.0);  glVertex2f(25.0, -15.0);  glVertex2f(-25.0, -15.0);  glVertex2f(-25.0, -10.0);  glEnd();  glFlush();  }  void myinit()  {  glClearColor(1.0, 1.0, 1.0, 0.0);  //glColor3f(1.0,0.0,0.0);  glPointSize(5.0);  glMatrixMode(GL\_PROJECTION);  gluOrtho2D(-40, 40.0, -40.0, 40.0);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(720, 560);  glutInitWindowPosition(0, 0);  glutCreateWindow("flag");  glutDisplayFunc(display);  myinit();  glutMainLoop();  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 2**  Draw 8X8 Chess Board |
| **Graph Plot (Picture)-** |
| **Code-**  #include<windows.h>  #include <GL/glut.h>  int x = -4, y = 4;  void display()  {  glLoadIdentity();  glBegin(GL\_QUADS);  for (int i = 0; i < 8; i++)  {  if (y == -4)  {  y = 4;  }  for (int j = 0; j < 8; j++)  {  if (x == 4)  {  x = -4;  }  if (i % 2 == 0)  {  if (j % 2 == 0)  {  glColor3f(1, 1, 1);  }  else  {  glColor3f(0, 0, 0);  }  }  else  {  if (j % 2 != 0)  {  glColor3f(1, 1, 1);  }  else  {  glColor3f(0, 0, 0);  }  }  glVertex2i(x, y);  glVertex2i(x, y - 1);  glVertex2i(x + 1, y - 1);  glVertex2i(x + 1, y);  x++;  }  y--;  }  glEnd();  glFlush();  }  void reshape(int w, int h)  {  glViewport(0, 0, w, h);  glMatrixMode(GL\_PROJECTION);  glLoadIdentity();  gluOrtho2D(-4.0, 4.0, -4.0, 4.0);  glMatrixMode(GL\_MODELVIEW);  }  void myinit()  {  glClearColor(1.0, 1.0, 0.0, 1.0);  }  int main(int argv, char\*\* argc)  {  glutInit(&argv, argc);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowPosition(400, 100);  glutInitWindowSize(500, 500);  glutCreateWindow("Chess");  glutDisplayFunc(display);  glutReshapeFunc(reshape);  myinit();  glutMainLoop();  } |
| **Output Screenshot (Full Screen)-** |

|  |
| --- |
| **Question- 3**  Create the batman logo given below- |
| **Graph Plot (Picture)-**  **(Not Needed)** |
| **Code-**  #include<Windows.h>  #include<GL/gl.h>  #include<Gl/glut.h>  #include<stdio.h>  #include<cmath>  using namespace std;  int m = 0;  void Line(int x, int y, int size, int num)  {  for (int a = x, b = y, c = 0; c < num; c++, a += size)  {  glBegin(GL\_POLYGON);  glVertex2i(a, b);  glVertex2i(a + size, b);  glVertex2i(a + size, b + size);  glVertex2i(a, b + size);  glEnd();  }  }  void vertical\_Line(int x, int y, int size, int num)  {  for (int a = x, b = y, c = 0; c < num; c++, b += size)  {  glBegin(GL\_POLYGON);  glVertex2i(a, b);  glVertex2i(a + size, b);  glVertex2i(a + size, b + size);  glVertex2i(a, b + size);  glEnd();  }  }  void temp(void)  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glPointSize(5);  int size = 25, block = 17;  int x = 9 \* size, y = 3 \* size;  glColor3ub(0, 0, 0);  Line(x, y, size, block);  x = 9 \* size, y = 4 \* size;  block;  for (int a = 0; a <= 5; a++)  {  glColor3ub(250, 250, 0);  Line(x, y, size, block);  glColor3ub(0, 0, 0);  Line(x - size, y, size, 1);  Line(x + (block \* size), y, size, 1);  x = x - size;  y = y + size;  block += 2;  }  x = 3 \* size, y = 10 \* size;  for (int a = 0; a < 7; a++)  {  glColor3ub(250, 250, 0);  Line(x, y, size, 29);  glColor3ub(0, 0, 0);  Line(x - size, y, size, 1);  glColor3ub(0, 0, 0);  Line(x + (29 \* size), y, size, 1);  y += size;  }  x = 4 \* size, y = 17 \* size, block = 27;  for (int a = 0; a <= 5; a++)  {  glColor3ub(250, 250, 0);  Line(x, y, size, block);  glColor3ub(0, 0, 0);  Line(x - size, y, size, 1);  Line(x + (block \* size), y, size, 1);  y = y + size;  x = x + size;  block -= 2;  }  x = 9 \* size;  glColor3ub(0, 0, 0);  Line(x, y, size, block + 2);  x = 4 \* size, y = 11 \* size;  block = 5;  for (int a = 0; a < 6; a++)  {  //glColor3ub(250, 250, 0);  vertical\_Line(x, y, size, block);  x += size;  y -= size;  block += 2;  }  block -= 2;  y += size;  vertical\_Line(x, y, size, block);  y -= size;  x += size;  block = 12;  vertical\_Line(x, y, size, block);  y = 20 \* size;  vertical\_Line(x, y, size, 1);  x += size;  y = 7 \* size;  block = 9;  for (int a = 0; a < 2; a++)  {  vertical\_Line(x, y, size, block);  x += size;  y += size;  block--;  }  block++;  vertical\_Line(x, y, size, block);  x += size;  y -= size;  block = 14;  vertical\_Line(x, y, size, block);  x += size;  y -= (2 \* size);  block = 15;  vertical\_Line(x, y, size, block);  x += size;  y -= (size);  block = 16;  vertical\_Line(x, y, size, block);  x += size;  y += (size);  block = 15;  vertical\_Line(x, y, size, block);  x += size;  y += (2 \* size);  block = 14;  vertical\_Line(x, y, size, block);  y = 6 \* size;  x = 24 \* size;  block = 16;  for (int a = 0; a <= 6; a++)  {  vertical\_Line(x, y, size, block);  x += size;  y += size;  block -= 2;  }  y = 6 \* size;  x = 23 \* size;  block = 16;  vertical\_Line(x, y, size, block);  y = 5 \* size;  x = 22 \* size;  block = 12;  vertical\_Line(x, y, size, block);  y = 21 \* size;  vertical\_Line(x, y, size, 1);  x -= size;  y -= (14 \* size);  vertical\_Line(x, y, size, 9);  x -= size;  y += size;  vertical\_Line(x, y, size, 9);  glFlush();  }  void display()  {  glClear(GL\_COLOR\_BUFFER\_BIT);  glPointSize(5.0);  glFlush();  }  void myinit()  {  glClearColor(1.0, 1.0, 1.0, 0.0);  glMatrixMode(GL\_MODELVIEW);  glLoadIdentity();  gluOrtho2D(0.0, 900.0, 0.0, 700.0);  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);  glutInitWindowSize(900, 700);  glutInitWindowPosition(200, 150);  glutCreateWindow("Batman");  glutDisplayFunc(display);  glutDisplayFunc(temp);  myinit();  glutMainLoop();  } |
| **Output Screenshot (Full Screen)-** |