**Computer Graphics**

**MidTerm**

**Performance Task**

*Instructions:*

* *The task must be shown during class time.*
* *Fill up all the sections in the tables below with the mentioned information.*
* *The completed task and this word file must be shown during class time by 14/7/2024 (Sunday).*
* *No late submission will be accepted.*

|  |
| --- |
| Question-1: Create the figure given below  Flag of Argentina | History, Design ...Flag of Bangladesh | Meaning, Colors ...  Flag of the United States of America ... |
| Graph: |
| Code:  #include <windows.h>  #include <GL/glut.h>  #include<math.h>  void code()  {  //USA  // box  glPointSize(5.0);  glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad  glColor3ub(0, 0, 102);  glVertex2f(1.0f, -0.61f); // Updated point  glVertex2f(4.0f, -0.61f);  glVertex2f(4.0f, -4.0f);  glVertex2f(1.0f, -4.0f);  glEnd();  //star  glBegin(GL\_POLYGON); // Each set of vertices forms a polygon  glColor3ub(255, 255, 255); // White color (RGB: 255, 255, 255)  glVertex2f(1.8f, -1.7f);  glVertex2f(1.4f, -1.9f);  glVertex2f(1.5f, -1.6f);  glVertex2f(1.2f, -1.41f);  glVertex2f(1.6f, -1.4f);  glVertex2f(1.8f, -1.0f);  glVertex2f(2.0f, -1.4f);  glVertex2f(2.4f, -1.4f);  glVertex2f(2.1f, -1.6f);  glVertex2f(2.1f, -2.0f);  glEnd();  glBegin(GL\_POLYGON); // Each set of vertices forms a polygon  glColor3ub(255, 255, 255); // White color (RGB: 255, 255, 255)  glVertex2f(3.3f, -1.6f);  glVertex2f(2.9383f, -1.82191f);  glVertex2f(3.0f, -1.5f);  glVertex2f(2.7f, -1.3f);  glVertex2f(3.1f, -1.3f);  glVertex2f(3.3f, -1.0f);  glVertex2f(3.5f, -1.31f);  glVertex2f(3.9f, -1.32f);  glVertex2f(3.5f, -1.5f);  glVertex2f(3.65229f, -1.83078f);  glEnd();  glBegin(GL\_POLYGON); // Each set of vertices forms a polygon  glColor3ub(255, 255, 255); // White color  glVertex2f(1.8f, -2.94f);  glVertex2f(1.3f, -3.2f);  glVertex2f(1.4f, -2.8f);  glVertex2f(1.11f, -2.51f);  glVertex2f(1.57f, -2.5f);  glVertex2f(1.8f, -2.06f);  glVertex2f(2.0f, -2.5f);  glVertex2f(2.38f, -2.5f);  glVertex2f(2.1f, -2.7f);  glVertex2f(2.2f, -3.2f);  glEnd();  glBegin(GL\_POLYGON); // Each set of vertices forms a polygon  glColor3ub(255, 255, 255); // White color (RGB: 255, 255, 255)  glVertex2f(3.26762f, -2.86631f);  glVertex2f(2.85127f, -3.22714f);  glVertex2f(2.93454f, -2.74835f);  glVertex2f(2.60147f, -2.49855f);  glVertex2f(3.0f, -2.5f);  glVertex2f(3.26762f, -2.06139f);  glVertex2f(3.5f, -2.5f);  glVertex2f(3.93376f, -2.50549f);  glVertex2f(3.64926f, -2.72753f);  glVertex2f(3.69664f, -3.13459f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2f(4.0f, -0.61f);  glVertex2f(4.0f, -0.83f);  glVertex2f(10.0f, -0.83f);  glVertex2f(10.0f, -0.61f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2f(4.0f, -1.3f);  glVertex2f(4.0f, -1.62f);  glVertex2f(10.0f, -1.61068f);  glVertex2f(10.0f, -1.31158f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2f(4.0f, -2.27f);  glVertex2f(4.0f, -2.61f);  glVertex2f(10.0f, -2.61f);  glVertex2f(10.0f, -2.27f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2f(4.0f, -3.18f);  glVertex2f(4.0f, -3.56f);  glVertex2f(10.0f, -3.56f);  glVertex2f(10.0f, -3.18f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2f(1.0f, -4.0f);  glVertex2f(1.0f, -4.46702f);  glVertex2f(10.0f, -4.47f);  glVertex2f(10.0f, -4.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2f(1.0f, -5.0f);  glVertex2f(1.0f, -5.41186f);  glVertex2f(10.0f, -5.41186f);  glVertex2f(10.0f, -5.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2f(10.0f, -6.55911f);  glVertex2f(10.0f, -6.0f);  glVertex2f(1.0f, -6.0f);  glVertex2f(1.0f, -6.55911f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(255, 0, 0);  glVertex2f(1.0f, -7.63391f);  glVertex2f(1.0f, -7.0f);  glVertex2f(10.0f, -7.0f);  glVertex2f(10.0f, -7.63391f);  glEnd();  // Argentina  // Argentina  // Argentina  glBegin(GL\_POLYGON);  glColor3ub(135, 206, 235); // Sky Blue color (RGB: 135, 206, 235)  glVertex2f(1.0f, 1.0f);  glVertex2f(1.0f, 3.0f);  glVertex2f(10.0f, 3.0f);  glVertex2f(10.0f, 1.0f);  glEnd();  glBegin(GL\_POLYGON);  glColor3ub(135, 206, 235); // Sky Blue color (RGB: 135, 206, 235)  glVertex2f(1.0f, 5.0f);  glVertex2f(10.0f, 5.0f);  glVertex2f(10.0f, 7.0f);  glVertex2f(1.0f, 7.0f);  glEnd();  //circle  glLineWidth(7.5);  glBegin(GL\_POLYGON); // Draw a Red 1x1 Square centered at origin  glColor3ub(255, 184, 28);  float centerX = 5.43034;  float centerY = 4.0331;  float radius = 0.7076;  for(int i = 0; i < 200; i++)  {  float pi = 3.1416;  float A = (i \* 2 \* pi) / 200;  float x = radius \* cos(A);  float y = radius \* sin(A);  glVertex2f(centerX + x, centerY + y);  }  glEnd();  // line  glLineWidth(1.0);  glBegin(GL\_LINES);  glColor3ub(184, 134, 11); // Dark Yellow color (RGB: 184, 134, 11)  glVertex2f(4.99296f, 4.26589f);  glVertex2f(5.37105f, 4.40338f);  glVertex2f(5.88663f, 4.1284f);  glVertex2f(5.6804f, 4.38619f);  glVertex2f(5.25075f, 3.68156f);  glVertex2f(5.66322f, 3.7f);  glEnd();  //BD  //BD  //BD  glPointSize(5.0);  glBegin(GL\_POLYGON); // Each set of 4 vertices form a quad  glColor3ub(0, 100, 0); // Dark green color (RGB: 0, 100, 0)  glVertex2f(-10.0f, 1.0f);  glVertex2f(-1.0f, 1.0f);  glVertex2f(-1.0f, 7.0f);  glVertex2f(-10.0f, 7.0f);  glEnd();  // BD CIRCLE  glLineWidth(7.5);  glBegin(GL\_POLYGON); // Draw a Red 1x1 Square centered at origin  glColor3ub(255, 0, 0); // Red color (RGB: 255, 0, 0)  float centerXx =-5.61339;  float centerYy = 4.01175;  float radiuss = 1.6018;  for(int i = 0; i < 200; i++)  {  float pi = 3.1416;  float A = (i \* 2 \* pi) / 200;  float x = radiuss \* cos(A);  float y = radiuss \* sin(A);  glVertex2f(centerXx + x, centerYy + y);  }  glEnd();  //sspp  //sspp  //Box  glPointSize(5.0);  glBegin(GL\_POLYGON); // Each set of 4 vertices forms a quad  glColor3ub(0, 60, 0); // Dark green color (RGB: 0, 128, 0)  glVertex2f(-10.0f, -7.0f);  glVertex2f(-1.0f, -7.0f);  glVertex2f(-1.0f, -5.0f);  glVertex2f(-10.0f, -5.0f);  glEnd();  glPointSize(5.0);  glBegin(GL\_POLYGON); // Each set of vertices forms a polygon  glColor3ub(220, 190, 0); // yellow color (RGB: 204, 204, 0)  glVertex2f(-10.0f, -5.0f);  glVertex2f(-10.0f, -3.0f);  glVertex2f(-1.0f, -3.0f);  glVertex2f(-1.0f, -5.0f);  glEnd();  glPointSize(5.0);  glBegin(GL\_POLYGON); // Each set of vertices forms a polygon  glColor3ub(255, 0, 0); // Red color (RGB: 255, 0, 0)  glVertex2f(-10.0f, -3.0f);  glVertex2f(-1.0f, -3.0f);  glVertex2f(-1.0f, -1.0f);  glVertex2f(-10.0f, -1.0f);  glEnd();  //star  glBegin(GL\_POLYGON); // Each set of vertices forms a polygon  glColor3ub(0, 0, 0); // Black color (RGB: 0, 0, 0)  glVertex2f(-5.16034f, -3.5817f);  glVertex2f(-4.5f, -3.61025f);  glVertex2f(-5.0f, -4.0f);  glVertex2f(-4.87821f, -4.58758f);  glVertex2f(-5.47573f, -4.14489f);  glVertex2f(-6.120483852393897f, -4.5816947386943685f);  glVertex2f(-6.0f, -4.0f);  glVertex2f(-6.5f, -3.59258f);  glVertex2f(-5.83617f, -3.5817f);  glVertex2f(-5.49825f, -3.08609f);  glEnd();  glFlush();  }  /\* 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);  glClear(GL\_COLOR\_BUFFER\_BIT);  code();  glEnd();  glFlush(); // Render now  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(500,500);  glutInitWindowPosition(10,10);  glutCreateWindow("LAB TASK");  glutDisplayFunc(display);  gluOrtho2D(-15,15,-15,15);  glutMainLoop();  return 0;  } |
| Output Screenshot: |

|  |
| --- |
| Question-2: Create the figure given below |
| Graph: |
| Code:  #include <windows.h>  #include <GL/glut.h>  #include<math.h>  void code()  {  //circle  glLineWidth(0.5);  glBegin(GL\_LINES); // Draw a Red 1x1 Square centered at origin  glColor3ub(105, 105, 105);  float centerX = -0.86;  float centerY = 0.98;  float radius =6.40802;  for(int i = 0; i < 200; i++)  {  float pi = 3.1416;  float A = (i \* 2 \* pi) / 200;  float x = radius \* cos(A);  float y = radius \* sin(A);  glVertex2f(centerX + x, centerY + y);  }  glEnd();  // cloud  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105); // Set the color to dark gray  // Existing lines  glVertex2f(-4.0242f, 5.4228f);  glVertex2f(-3.4192f, 5.81f);  glVertex2f(-2.8384f, 6.1004f);  glVertex2f(-1.1928f, 6.9716f);  glVertex2f(-6.9766f, 2.7366f);  glVertex2f(-4.6534f, 4.6726f);  glVertex2f(-4.6534f, 4.6726f);  glVertex2f(-4.0f, 4.4548f);  glVertex2f(-4.0f, 4.4548f);  glVertex2f(-1.7978f, 5.326f);  glVertex2f(-1.7978f, 5.326f);  glVertex2f(-0.01217f, 4.29792f);  glVertex2f(-0.01217f, 4.29792f);  glVertex2f(1.1788f, 4.4064f);  glVertex2f(1.1788f, 4.4064f);  glVertex2f(2.8652f, 2.88588f);  glVertex2f(1.2272f, 2.785f);  glVertex2f(0.7916f, 3.5836f);  glVertex2f(0.7916f, 3.5836f);  glVertex2f(1.1788f, 4.4064f);  glVertex2f(-1.2896f, 2.906f);  glVertex2f(-0.8782f, 3.39f);  glVertex2f(-0.8782f, 3.39f);  glVertex2f(-1.7978f, 4.479f);  glVertex2f(-1.7978f, 4.479f);  glVertex2f(-1.7978f, 5.326f);  glVertex2f(-4.1936f, 3.1964f);  glVertex2f(-4.7018f, 3.4384f);  glVertex2f(-4.7018f, 3.4384f);  glVertex2f(-4.6534f, 4.6726f);  glVertex2f(-6.3474f, 0.607f);  glVertex2f(-5.936f, 2.2768f);  glVertex2f(-5.936f, 2.2768f);  glVertex2f(-6.31f, 3.29f);  glVertex2f(1.96f, 3.7f);  glVertex2f(3.4294f, 5.2292f);  glVertex2f(3.4294f, 5.2292f);  glVertex2f(3.4294f, 5.2292f);  glVertex2f(2.6066f, 1.2604f);  glVertex2f(4.0f, 2.0f);  glVertex2f(4.0f, 2.0f);  glVertex2f(3.6714f, 3.1964f);  glVertex2f(3.6714f, 3.1964f);  glVertex2f(4.0f, 4.0f);  glVertex2f(4.0f, 4.0f);  glVertex2f(3.4294f, 5.2292f);  glVertex2f(4.6636f, 2.422f);  glVertex2f(5.0266f, 3.3658f);  glVertex2f(4.93f, 3.72f);  glVertex2f(3.4294f, 5.2292f);  glVertex2f(-2.959f, 1.285f);  glVertex2f(-3.226f, 2.011f);  glVertex2f(-3.226f, 2.011f);  glVertex2f(-2.693f, 2.712f);  glVertex2f(-2.693f, 2.712f);  glVertex2f(-2.984f, 3.196f);  glVertex2f(-2.715f, 1.503f);  glVertex2f(-2.841f, 2.054f);  glVertex2f(-2.841f, 2.054f);  glVertex2f(-2.379f, 2.572f);  glVertex2f(-2.379f, 2.572f);  glVertex2f(-2.547f, 3.161f);  glEnd();  //star  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105); // Set the color to dark gray  glVertex2f(-0.03127f, 7.13357f);  glVertex2f(0.30074f, 6.3992f);  glVertex2f(0.30074f, 6.3992f);  glVertex2f(0.81585f, 6.42112f);  glVertex2f(0.81585f, 6.42112f);  glVertex2f(0.4761f, 6.10328f);  glVertex2f(0.4761f, 6.10328f);  glVertex2f(0.519f, 5.4f);  glVertex2f(0.519f, 5.4f);  glVertex2f(0.0166f, 5.8f);  glVertex2f(0.0166f, 5.8f);  glVertex2f(-0.4859f, 5.37f);  glVertex2f(-0.4859f, 5.37f);  glVertex2f(-0.49934f, 6.03752f);  glVertex2f(-0.49934f, 6.03752f);  glVertex2f(-0.89389f, 6.3992f);  glVertex2f(-0.89389f, 6.3992f);  glVertex2f(-0.32398f, 6.41016f);  glVertex2f(-0.32398f, 6.41016f);  glVertex2f(-0.03127f, 7.13357f);  glEnd();  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105); // Set the color to dark gray  glVertex2f(1.780f, 6.564f);  glVertex2f(2.000f, 6.000f);  glVertex2f(2.000f, 6.000f);  glVertex2f(2.569f, 6.016f);  glVertex2f(2.569f, 6.016f);  glVertex2f(2.219f, 5.676f);  glVertex2f(2.219f, 5.676f);  glVertex2f(2.306f, 5.161f);  glVertex2f(2.306f, 5.161f);  glVertex2f(1.846f, 5.566f);  glVertex2f(1.846f, 5.566f);  glVertex2f(1.441f, 5.161f);  glVertex2f(1.441f, 5.161f);  glVertex2f(1.386f, 5.665f);  glVertex2f(1.386f, 5.665f);  glVertex2f(1.000f, 6.000f);  glVertex2f(1.000f, 6.000f);  glVertex2f(1.517f, 5.994f);  glVertex2f(1.517f, 5.994f);  glVertex2f(1.780f, 6.564f);  glEnd();  // House  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105); // Set the color to dark gray  glVertex2f(2.052f, -0.162f);  glVertex2f(0.947f, 0.101f);  glVertex2f(0.947f, 0.101f);  glVertex2f(-0.900f, 0.200f);  glVertex2f(-0.900f, 0.200f);  glVertex2f(-1.372f, 0.301f);  glVertex2f(-1.372f, 0.301f);  glVertex2f(-2.465f, 0.212f);  glVertex2f(-2.465f, 0.212f);  glVertex2f(-3.200f, 0.011f);  glVertex2f(-3.200f, 0.011f);  glVertex2f(-3.851f, -0.201f);  glVertex2f(0.188f, 2.285f);  glVertex2f(0.947f, 0.101f);  glVertex2f(0.188f, 2.285f);  glVertex2f(-1.729f, 2.219f);  glVertex2f(-1.729f, 2.219f);  glVertex2f(-0.900f, 0.200f);  glVertex2f(-1.372f, 0.301f);  glVertex2f(-1.729f, 2.219f);  glVertex2f(-1.729f, 2.219f);  glVertex2f(-2.465f, 0.212f);  glVertex2f(-3.200f, 0.011f);  glVertex2f(-1.729f, 2.219f);  glVertex2f(-1.088f, 1.796f);  glVertex2f(-0.141f, 1.796f);  glVertex2f(-0.796f, 1.426f);  glVertex2f(-0.248f, 1.426f);  glEnd();  // Tree  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105);  glVertex2f(3.400f, -3.661f);  glVertex2f(3.400f, 0.895f);  glVertex2f(2.321f, -2.681f);  glVertex2f(3.400f, -1.566f);  glVertex2f(3.400f, -1.566f);  glVertex2f(4.378f, -2.508f);  glVertex2f(2.552f, -1.508f);  glVertex2f(3.400f, -0.700f);  glVertex2f(3.400f, -0.700f);  glVertex2f(4.205f, -1.470f);  glVertex2f(2.744f, -0.527f);  glVertex2f(3.400f, 0.113f);  glVertex2f(3.400f, 0.113f);  glVertex2f(4.023f, -0.574f);  glVertex2f(2.821f, 0.222f);  glVertex2f(3.400f, 0.895f);  glVertex2f(3.400f, 0.895f);  glVertex2f(3.974f, 0.338f);  glEnd();  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105);  glVertex2f(1.167f, -5.007f);  glVertex2f(1.167f, -0.508f);  glVertex2f(0.148f, -4.123f);  glVertex2f(1.167f, -3.000f);  glVertex2f(1.167f, -3.000f);  glVertex2f(2.206f, -3.930f);  glVertex2f(0.437f, -2.873f);  glVertex2f(1.167f, -2.000f);  glVertex2f(1.167f, -2.000f);  glVertex2f(1.975f, -2.758f);  glVertex2f(0.514f, -1.892f);  glVertex2f(1.167f, -1.300f);  glVertex2f(1.167f, -1.300f);  glVertex2f(1.783f, -1.854f);  glVertex2f(0.571f, -0.989f);  glVertex2f(1.167f, -0.508f);  glVertex2f(1.167f, -0.508f);  glVertex2f(1.725f, -0.912f);  glEnd();  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105);  glVertex2f(-1.000f, -5.256f);  glVertex2f(-1.000f, -1.000f);  glVertex2f(-1.991f, -4.400f);  glVertex2f(-1.000f, -3.300f);  glVertex2f(-1.000f, -3.300f);  glVertex2f(-0.071f, -4.365f);  glVertex2f(-1.843f, -3.148f);  glVertex2f(-1.000f, -2.300f);  glVertex2f(-1.000f, -2.300f);  glVertex2f(-0.240f, -3.060f);  glVertex2f(-1.711f, -2.292f);  glVertex2f(-1.000f, -1.700f);  glVertex2f(-1.000f, -1.700f);  glVertex2f(-0.416f, -2.173f);  glVertex2f(-1.522f, -1.393f);  glVertex2f(-1.000f, -1.000f);  glVertex2f(-1.000f, -1.000f);  glVertex2f(-0.452f, -1.448f);  glEnd();  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105);  glVertex2f(-3.000f, -5.000f);  glVertex2f(-3.000f, -0.500f);  glVertex2f(-4.000f, -4.000f);  glVertex2f(-3.000f, -2.900f);  glVertex2f(-3.000f, -2.900f);  glVertex2f(-2.000f, -4.000f);  glVertex2f(-3.824f, -2.751f);  glVertex2f(-3.000f, -2.000f);  glVertex2f(-3.000f, -2.000f);  glVertex2f(-2.314f, -2.667f);  glVertex2f(-4.000f, -2.000f);  glVertex2f(-3.000f, -1.000f);  glVertex2f(-3.000f, -1.000f);  glVertex2f(-2.195f, -2.000f);  glVertex2f(-3.551f, -0.989f);  glVertex2f(-3.000f, -0.500f);  glVertex2f(-3.000f, -0.500f);  glVertex2f(-2.566f, -0.948f);  glEnd();  glLineWidth(3.0);  glBegin(GL\_LINES);  glColor3ub(105, 105, 105);  glVertex2f(-5.300f, -3.422f);  glVertex2f(-5.300f, 1.233f);  glVertex2f(-6.340f, -2.080f);  glVertex2f(-5.300f, -1.200f);  glVertex2f(-5.300f, -1.200f);  glVertex2f(-4.327f, -2.206f);  glVertex2f(-6.151f, -0.927f);  glVertex2f(-5.300f, -0.200f);  glVertex2f(-5.300f, -0.200f);  glVertex2f(-4.579f, -0.885f);  glVertex2f(-6.000f, 0.000f);  glVertex2f(-5.300f, 0.500f);  glVertex2f(-5.300f, 0.500f);  glVertex2f(-4.600f, -0.067f);  glVertex2f(-5.774f, 0.793f);  glVertex2f(-5.300f, 1.233f);  glVertex2f(-5.300f, 1.233f);  glVertex2f(-4.888f, 0.840f);  glEnd();  glFlush();  }  /\* 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);  glClear(GL\_COLOR\_BUFFER\_BIT);  code();  glEnd();  glFlush(); // Render now  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutInitWindowSize(640,480);  glutInitWindowPosition(10,10);  glutCreateWindow("LAB TASK");  glutDisplayFunc(display);  gluOrtho2D(-15,15,-15,15);  glutMainLoop();  return 0;  } |
| Output Screenshot: |