|  |
| --- |
| National Flag of Brazil: |
| Graph: |
| #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  void Brazil()  {  glColor3ub(11, 121, 26);  glBegin(GL\_POLYGON);  glVertex2f(-5, 3);  glVertex2f(5,3);  glVertex2f(5,-3);  glVertex2f(-5,-3);  glEnd();  }  void central\_rectanlge()  {  glColor3ub(255, 255, 0);  glBegin(GL\_POLYGON);  glVertex2f(-3,0);  glVertex2f(0,2);  glVertex2f(3,0);  glVertex2f(0,-2);  glEnd();  }  void circle()  {  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(34, 13, 120);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=1.24;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x,y );  }  glEnd();  }  void central\_border()  {  glColor3ub(255, 255, 255);  glBegin(GL\_POLYGON);  glVertex2f(-0.706, 0.932);  glVertex2f(-0.2, 0.756);  glVertex2f(0.13, 0.558);  glVertex2f(0.482, 0.338);  glVertex2f(0.812, 0.008);  glVertex2f(1.186, -0.498);  glVertex2f(-0.98265, 0.633602730);  glVertex2f(-0.42, 0.514);  glVertex2f(0.306, 0.03);  glVertex2f(0.306, 0.03);  glVertex2f(0.966, -0.784);  glEnd();  }  void display()  {  glClearColor(0.0f, 0.0f, 0.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  Brazil();  central\_rectanlge(); // Clear the color buffer (background)  circle();  central\_border();  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");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  glutInitWindowSize(320, 320);// Set the window's initial width & height  glutDisplayFunc(display);// Register display callback handler for window re-paint  gluOrtho2D(-8,8,-6,6);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| Output |

|  |
| --- |
| National Flag of Argentina: |
| Graph |
| #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  void Argentina()  {  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(8, 4.8);  glVertex2f(9.2, 4.8);  glVertex2f(9.2,4);  glVertex2f(8,4);  glEnd();  //1st  glColor3ub(4, 205, 246);  glBegin(GL\_POLYGON);  glVertex2f(8,4.8);  glVertex2f(9.2, 4.8);  glVertex2f(9.2,4.5);  glVertex2f(8,4.5);  glEnd();  //2nd  glColor3ub(255, 255, 255);  glBegin(GL\_POLYGON);  glVertex2f(8, 4.5);  glVertex2f(9.2,4.5);  glVertex2f(9.2,4.2);  glVertex2f(8,4.2);  glEnd();  glColor3ub(4, 205, 246);  glBegin(GL\_POLYGON);  glVertex2f(8,4.2);  glVertex2f(9.2,4.2);  glVertex2f(9.2,4);  glVertex2f(8,4);  glEnd();  glColor3ub(246, 220, 4);  glLineWidth(3.5);  glBegin(GL\_LINES);  glVertex2f(8.59981,4.40771836635);  glVertex2f(8.6, 4.45);  glVertex2f(8.6324329, 4.3999);  glVertex2f(8.6512357958138, 4.4421526878665);  glVertex2f(8.64993, 4.38527);  glVertex2f(8.6838927928754, 4.4114575186483);  glVertex2f(8.66174103,4.36658895);  glVertex2f(8.6993182840054, 4.3780356212);  glVertex2f(8.666988589,4.3394400237914);  glVertex2f(8.706174057841, 4.339471893375);  glVertex2f(8.66350, 4.3191527);  glVertex2f(8.7, 4.3);  glVertex2f(8.6520764,4.29832260);  glVertex2f(8.6809110978499, 4.2727414170542);  glVertex2f(8.6369,4.284);  glVertex2f(8.6564696975332, 4.2503468335129);  glVertex2f(8.6809110978499, 4.2727414170542);  glVertex2f(8.6341884325677, 4.2426340879479);  glVertex2f(8.5998174920471, 4.2733582162671);  glVertex2f(8.6016235068488, 4.2383492293006);  glVertex2f(8.5723654730923, 4.2795600332344);  glVertex2f(8.5604888638355, 4.2417771162184);  glVertex2f(8.5520377640412,4.2933126618853);  glVertex2f(8.5313518250344, 4.2572026073484);  glVertex2f(8.5382045033894,4.3137613962013);  glVertex2f(8.4987868993156, 4.2923384482556);  glVertex2f(8.53826436445737,4.33396235789172);  glVertex2f(8.4880963346835, 4.3396192534263);  glVertex2f(8.53826436445737,4.33396235789172);  glVertex2f(8.4880963346835, 4.3396192534263);  glVertex2f(8.5369276831437,4.3641610434651);  glVertex2f(8.5005008427745, 4.3874623102239);  glVertex2f(8.5514444957687,4.3871560531017);  glVertex2f(8.52278210774, 4.4131714621072);  glVertex2f(8.5736729856362,4.402422273632);  glVertex2f(8.5553470334589, 4.4388806139905);  glEnd();  }  void circle()  {  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(246, 220, 4);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=0.070;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+8.599,y+4.34 );  }  glEnd();  }  void display() {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  Argentina(); // Clear the color buffer (background)  circle();  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");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  glutInitWindowSize(320, 320);// Set the window's initial width & height  glutDisplayFunc(display);// Register display callback handler for window re-paint  gluOrtho2D(7,10,3,6);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| Output |

|  |
| --- |
| National Flag of Greece: |
| Graph |
| #include <windows.h>  #include <GL/glut.h>  #include <math.h>  void flag\_2()  {  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(-9,-1);  glVertex2f(-1,-1);  glVertex2f(-1,-5);  glVertex2f(-9,-5);  glEnd();  //whitemidddle  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-9,-2.3);  glVertex2f(-9,-1.9);  glVertex2f(-8,-1.9);  glVertex2f(-8,-1);  glVertex2f(-7.5,-1);  glVertex2f(-7.5,-1.9);  glVertex2f(-6.6,-1.9);  glVertex2f(-6.6,-2.3);  glVertex2f(-7.5,-2.3);  glVertex2f(-7.5,-3.19);  glVertex2f(-8,-3.19);  glVertex2f(-8,-2.3);  glEnd();  //blue1  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-9,-1.9);  glVertex2f(-9,-1);  glVertex2f(-8,-1);  glVertex2f(-8,-1.9);  glEnd();  //blue2  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-7.5,-1.9);  glVertex2f(-7.5,-1);  glVertex2f(-6.6,-1);  glVertex2f(-6.6,-1.5);  glVertex2f(-6.6,-1.9);  glEnd();  //blue3  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-7.5,-3.19);  glVertex2f(-7.5,-2.3);  glVertex2f(-6.6,-2.3);  glVertex2f(-6.6,-2.8);  glVertex2f(-6.6,-3.2);  glEnd();  //blue4  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-9,-3.19);  glVertex2f(-9,-2.3);  glVertex2f(-8,-2.3);  glVertex2f(-8,-3.19);  glEnd();  //allLine  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-6.6,-1.5);  glVertex2f(-1,-1.5);  glVertex2f(-1,-1.9);  glVertex2f(-6.6,-1.9);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-6.6,-2.3);  glVertex2f(-1,-2.3);  glVertex2f(-1,-2.8);  glVertex2f(-6.6,-2.8);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-9,-3.2);  glVertex2f(-1,-3.2);  glVertex2f(-1,-3.65);  glVertex2f(-9,-3.65);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-9,-4.1);  glVertex2f(-1,-1.5);  glVertex2f(-1,-4.5);  glVertex2f(-9,-4.5);  glEnd();  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-6.6,-1);  glVertex2f(-1,-1);  glVertex2f(-1,-1.5);  glVertex2f(-6.6,-1.5);  glEnd();  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-6.6,-1.9);  glVertex2f(-1,-1.9);  glVertex2f(-1,-2.3);  glVertex2f(-6.6,-2.3);  glEnd();  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-6.6,-2.8);  glVertex2f(-1,-2.8);  glVertex2f(-1,-3.2);  glVertex2f(-6.6,-3.2);  glEnd();  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-9,-3.65);  glVertex2f(-1,-3.65);  glVertex2f(-1,-4.1);  glVertex2f(-9,-4.1);  glEnd();  glColor3ub(88,114,184 );  glBegin(GL\_POLYGON);  glVertex2f(-9,-4.5);  glVertex2f(-1,-4.5);  glVertex2f(-1,-5);  glVertex2f(-9,-5);  glEnd();  }  void display()  {  glClearColor(0,0,0,0);  glClear(GL\_COLOR\_BUFFER\_BIT);  flag\_2();  glFlush();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutCreateWindow("Flag\_1");  glutInitWindowSize(320, 320);  glutDisplayFunc(display);  gluOrtho2D(-20,20,-10,10);  glutMainLoop();  return 0;  }Output |

|  |
| --- |
| FLAG OF YUGOSLAVIA |
| Graph |
| #include <GL/glut.h>  #include <math.h>  void flag\_2()  {  glColor3ub(14, 15, 221);  glBegin(GL\_POLYGON);  glVertex2f(-10,-1.8);  glVertex2f(-2,-1.8);  glVertex2f(-2,-2.8);  glVertex2f(-10,-2.8);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-2,-2.8);  glVertex2f(-10,-2.8);  glVertex2f(-10,-5.2);  glVertex2f(-2,-5.2);  glEnd();  glColor3ub(14, 15, 221);  glBegin(GL\_POLYGON);  glVertex2f(-10,-5.2);  glVertex2f(-2,-5.2);  glVertex2f(-2,-6.2);  glVertex2f(-10,-6.2);  glEnd();  glColor3ub(14, 15, 221);  glLineWidth(8);  glBegin(GL\_LINES);  glVertex2f(-7.4,-3.5);  glVertex2f(-4.6,-3.5);  glVertex2f(-4.6,-3.5);  glVertex2f(-6,-5);  glVertex2f(-6,-5);  glVertex2f(-7.4,-3.5);  glVertex2f(-6,-3);  glVertex2f(-4.6,-4.5);  glVertex2f(-4.6,-4.5);  glVertex2f(-7.4,-4.5);  glVertex2f(-7.4,-4.5);  glVertex2f(-6,-3);  glEnd();  }  void display()  {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f);  glClear(GL\_COLOR\_BUFFER\_BIT);  flag\_3();  glFlush();  }  int main(int argc, char\*\* argv) {  glutInit(&argc, argv);  glutCreateWindow("Flag\_2");  glutInitWindowSize(320, 320);  glutDisplayFunc(display);  gluOrtho2D(-12,12,-9,9);  glutMainLoop();  return 0;  } |
| Output |

|  |
| --- |
| National Flag of USA |
| Graph : British East India Company Flag |
| #include <windows.h>  #include <GL/glut.h>  #include <math.h>  void flag()  {  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(1,-1);  glVertex2f(10,-1);  glVertex2f(10,-6.18);  glVertex2f(1,-6.18);  glEnd();  //blue  glColor3ub(17,20,116);  glBegin(GL\_POLYGON);  glVertex2f(1,-3.8);  glVertex2f(1,-1);  glVertex2f(4.2,-1);  glVertex2f(4.2,-3.8);  glEnd();  //white1  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(2.6,-1.1);  glVertex2f(2.6,-1.2);  glVertex2f(2.7,-1.2);  glVertex2f(2.6,-1.3);  glVertex2f(2.7,-1.4);  glVertex2f(2.6,-1.3);  glVertex2f(2.5,-1.4);  glVertex2f(2.5,-1.3);  glVertex2f(2.4,-1.2);  glVertex2f(2.5,-1.2);  glEnd();  //white2  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(3.1,-1.2);  glVertex2f(3.1,-1.4);  glVertex2f(3.2,-1.4);  glVertex2f(3.1,-1.4);  glVertex2f(3.2,-1.6);  glVertex2f(3.1,-1.5);  glVertex2f(3,-1.6);  glVertex2f(3,-1.4);  glVertex2f(2.9,-1.4);  glVertex2f(3,-1.4);  glEnd();  //white3  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(3.4,-1.6);  glVertex2f(3.5,-1.8);  glVertex2f(3.6,-1.8);  glVertex2f(3.5,-1.8);  glVertex2f(3.6,-2);  glVertex2f(3.4,-1.9);  glVertex2f(3.3,-2);  glVertex2f(3.4,-1.8);  glVertex2f(3.2,-1.8);  glVertex2f(3.4,-1.8);  glEnd();  //white4  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(3.7,-2.2);  glVertex2f(3.7,-2.3);  glVertex2f(3.8,-2.3);  glVertex2f(3.7,-2.3);  glVertex2f(3.8,-2.5);  glVertex2f(3.7,-2.4);  glVertex2f(3.6,-2.5);  glVertex2f(3.6,-2.3);  glVertex2f(3.5,-2.3);  glVertex2f(3.6,-2.3);  glEnd();  //white5  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(3.6,-2.6);  glVertex2f(3.6,-2.8);  glVertex2f(3.8,-2.8);  glVertex2f(3.6,-2.8);  glVertex2f(3.7,-3);  glVertex2f(3.6,-2.9);  glVertex2f(3.5,-3);  glVertex2f(3.5,-2.8);  glVertex2f(3.4,-2.8);  glVertex2f(3.5,-2.8);  glEnd();  //white6  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(3.2,-3.1);  glVertex2f(3.3,-3.2);  glVertex2f(3.4,-3.2);  glVertex2f(3.3,-3.3);  glVertex2f(3.3,-3.4);  glVertex2f(3.2,-3.3);  glVertex2f(3.1,-3.4);  glVertex2f(3.2,-3.3);  glVertex2f(3.1,-3.2);  glVertex2f(3.2,-3.2);  glEnd();  //white7  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(2.7,-3.3);  glVertex2f(2.7,-3.4);  glVertex2f(2.9,-3.4);  glVertex2f(2.8,-3.5);  glVertex2f(2.8,-3.6);  glVertex2f(2.7,-3.5);  glVertex2f(2.6,-3.6);  glVertex2f(2.6,-3.5);  glVertex2f(2.5,-3.5);  glVertex2f(2.7,-3.4);  glEnd();  //white8  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(2.1,-3.3);  glVertex2f(2.2,-3.4);  glVertex2f(2.3,-3.4);  glVertex2f(2.2,-3.5);  glVertex2f(2.2,-3.6);  glVertex2f(2.1,-3.5);  glVertex2f(2,-3.6);  glVertex2f(2.1,-3.5);  glVertex2f(2,-3.4);  glVertex2f(2.1,-3.4);  glEnd();  //white9  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(1.7,-3.1);  glVertex2f(1.7,-3.2);  glVertex2f(1.8,-3.2);  glVertex2f(1.7,-3.2);  glVertex2f(1.8,-3.3);  glVertex2f(1.7,-3.3);  glVertex2f(1.6,-3.3);  glVertex2f(1.6,-3.2);  glVertex2f(1.5,-3.2);  glVertex2f(1.7,-3.2);  glEnd();  //white10  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(1.5,-2.5);  glVertex2f(1.5,-2.7);  glVertex2f(1.6,-2.7);  glVertex2f(1.5,-2.7);  glVertex2f(1.6,-2.9);  glVertex2f(1.5,-2.8);  glVertex2f(1.4,-2.9);  glVertex2f(1.4,-2.7);  glVertex2f(1.3,-2.7);  glVertex2f(1.4,-2.7);  glEnd();  //white11  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(1.4,-2);  glVertex2f(1.5,-2.1);  glVertex2f(1.6,-2.1);  glVertex2f(1.5,-2.2);  glVertex2f(1.5,-2.3);  glVertex2f(1.4,-2.2);  glVertex2f(1.3,-2.3);  glVertex2f(1.4,-2.2);  glVertex2f(1.3,-2.1);  glVertex2f(1.4,-2.1);  glEnd();  //white12  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(1.6,-1.6);  glVertex2f(1.6,-1.7);  glVertex2f(1.8,-1.7);  glVertex2f(1.7,-1.8);  glVertex2f(1.7,-1.9);  glVertex2f(1.6,-1.8);  glVertex2f(1.5,-1.9);  glVertex2f(1.6,-1.8);  glVertex2f(1.5,-1.7);  glVertex2f(1.6,-1.7);  glEnd();  //white13  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(2,-1.2);  glVertex2f(2.1,-1.3);  glVertex2f(2.2,-1.3);  glVertex2f(2.1,-1.4);  glVertex2f(2.1,-1.5);  glVertex2f(2,-1.4);  glVertex2f(1.9,-1.5);  glVertex2f(2,-1.3);  glVertex2f(1.9,-1.3);  glVertex2f(2,-1.3);  glEnd();  //all line  //red1  glColor3ub(245,49,26);  glBegin(GL\_POLYGON);  glVertex2f(4.2,-1.4);  glVertex2f(4.2,-1);  glVertex2f(10,-1);  glVertex2f(10,-1.4);  glEnd();  //red2  glColor3ub(245,49,26);  glBegin(GL\_POLYGON);  glVertex2f(4.2,-2.2);  glVertex2f(4.2,-1.8);  glVertex2f(10,-1.8);  glVertex2f(10,-2.2);  glEnd();  //red3  glColor3ub(245,49,26);  glBegin(GL\_POLYGON);  glVertex2f(4.2,-3);  glVertex2f(4.2,-2.6);  glVertex2f(10,-2.6);  glVertex2f(10,-3);  glEnd();  //red4  glColor3ub(245,49,26);  glBegin(GL\_POLYGON);  glVertex2f(4.2,-3.8);  glVertex2f(4.2,-3.4);  glVertex2f(10,-3.4);  glVertex2f(10,-3.8);  glEnd();  //red5  glColor3ub(245,49,26 );  glBegin(GL\_POLYGON);  glVertex2f(1,-4.6);  glVertex2f(1,-4.2);  glVertex2f(10,-4.2);  glVertex2f(10,-4.6);  glEnd();  //red6  glColor3ub(245,49,26 );  glBegin(GL\_POLYGON);  glVertex2f(1,-5.5);  glVertex2f(1,-5);  glVertex2f(10,-5);  glVertex2f(10,-5.5);  glEnd();  //red7  glColor3ub(245,49,26 );  glBegin(GL\_POLYGON);  glVertex2f(1,-6.18);  glVertex2f(1,-5.8);  glVertex2f(10,-5.8);  glVertex2f(10,-6.18);  glEnd();  //white1  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(4.2,-1.8);  glVertex2f(4.2,-1.4);  glVertex2f(10,-1.4);  glVertex2f(10,-1.8);  glEnd();  //white2  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(4.2,-2.6);  glVertex2f(4.2,-2.2);  glVertex2f(10,-2.2);  glVertex2f(10,-2.6);  glEnd();  //white3  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(4.2,-3.4);  glVertex2f(4.2,-3);  glVertex2f(10,-3);  glVertex2f(10,-3.4);  glEnd();  //white4  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(1,-4.2);  glVertex2f(1,-3.8);  glVertex2f(10,-3.8);  glVertex2f(10,-4.2);  glEnd();  //white5  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(1,-5);  glVertex2f(1,-4.6);  glVertex2f(10,-4.6);  glVertex2f(10,-5);  glEnd();  //white6  glColor3ub(255,255,255 );  glBegin(GL\_POLYGON);  glVertex2f(1,-5.8);  glVertex2f(1,-5.5);  glVertex2f(10,-5.5);  glVertex2f(10,-5.8);  glEnd();  }  void display()  {  glClearColor(0,0,0,0);  glClear(GL\_COLOR\_BUFFER\_BIT);  flag();  glFlush();  }  int main(int argc, char\*\* argv)  {  glutInit(&argc, argv);  glutCreateWindow("Flag");  glutInitWindowSize(320,320);  glutDisplayFunc(display);  gluOrtho2D(-20,20,-10,10);  glutMainLoop();  return 0;  } |
| Output |

|  |
| --- |
| Name: SAMIA |
| Graph |
| #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  void myName()  {  glColor3ub(0,0,0);  glLineWidth(2);  glBegin(GL\_LINES);    glVertex2f(6,8);  glVertex2f(3.186884802596, 7.9988912925906);  glVertex2f(3.186884802596, 7.9988912925906);  glVertex2f(3.1659437533802, 5.6325527312061);  glVertex2f(3.1659437533802, 5.6325527312061);  glVertex2f(5.9720443482964, 5.6116116819903);  glVertex2f(5.9720443482964, 5.6116116819903);  glVertex2f(5.9929853975122, 3.1405678745268);  glVertex2f(5.9929853975122, 3.1405678745268);  glVertex2f(3.1240616549486, 3.1824499729584);  glVertex2f(6.8096863169281, 3.119626825311);  glVertex2f(8, 8);  glVertex2f(8, 8);  glVertex2f(9.1760248783126, 3.2243320713899);  glVertex2f(7.38525410,5.4794968466);  glVertex2f(8.624299,5.4648085);  //U  glVertex2f(9.5948458626284, 3.1824499729584);  glVertex2f(9.5529637641969, 7.9988912925906);  glVertex2f(9.5529637641969, 7.9988912925906);  glVertex2f(11.0188372093023, 6.0304326663061);  glVertex2f(11.0188372093023, 6.0304326663061);  glVertex2f(12.2753001622499, 8.0826554894538);  glVertex2f(12.2753001622499, 8.0826554894538);  glVertex2f(12.4009464575446, 3.1824499729584);  glVertex2f(12.6941211465657, 8.061714440238);  glVertex2f(14.4112871822607, 8.040773391022);  glVertex2f(13.57,8.05);  glVertex2f(13.6364683612764, 3.2662141698215);  glVertex2f(12.7778853434289, 3.2452731206057);  glVertex2f(14.4112871822607, 3.2662141698215);  glVertex2f(14.8091671173607, 3.2662141698215);  glVertex2f(16, 8);  glVertex2f(16, 8);  glVertex2f(17.4896214169821, 3.2033910221742);  glVertex2f(15.36,5.46665794);  glVertex2f(16.777641,5.49597);  glEnd();  }  void display()  {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  myName();  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");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  glutInitWindowSize(320, 320);// Set the window's initial width & height  glutDisplayFunc(display);// Register display callback handler for window re-paint  gluOrtho2D(-20,20,-10,10);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| Output |

|  |
| --- |
| Text: AIUB |
| Graph: |
| #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  void AIUB()  {  glColor3ub(0,0,0);  glLineWidth(2);  glBegin(GL\_LINES);  //T  glVertex2f(2.48, -0.04);  glVertex2f(3.82, 4.02);  glVertex2f(3.82, 4.02);  glVertex2f(5, 0);  glVertex2f(3.031154, 1.629914);  glVertex2f(4.52564,1.616);  glVertex2f(5.4, 3.98);  glVertex2f(7, 4);  glVertex2f(6.199875,3.9899);  glVertex2f(6.24,0);  glVertex2f(5.5,0);  glVertex2f(7,0);  glVertex2f(7.64, 4.02);  glVertex2f(7.616527, 0);  glVertex2f(7.616527, 0);  glVertex2f(9.74,0);  glVertex2f(9.74,0);  glVertex2f(9.72, 3.98);  //U  glVertex2f(10.2, 3.98);  glVertex2f(12.3909094796685, 3.9845663903807);  glVertex2f(12.3909094796685, 3.9845663903807);  glVertex2f(12.4, 2.02);  glVertex2f(12.4, 2.02);  glVertex2f(10.2098,2.0298);  glVertex2f(12.4, 2.02);  glVertex2f(12.42, 0);  glVertex2f(12.42, 0);  glVertex2f(10.22,0);  glVertex2f(10.22,0);  glVertex2f(10.22, 3.98);  glEnd();  }  void display()  {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  AIUB();  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");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  glutInitWindowSize(320, 320);// Set the window's initial width & height  glutDisplayFunc(display);// Register display callback handler for window re-paint  gluOrtho2D(-20,20,-10,10);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| Output: |

|  |
| --- |
| Text:NEPTUNE |
| Graph: |
| Code:  Code:  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  void myName()  {  //N  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(-7,2);  glVertex2f(-5,2);  glVertex2f(-5,0);  glVertex2f(-7,0);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-6.5,2);  glVertex2f(-5.5,1);  glVertex2f(-5.5,2);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-5.5,0);  glVertex2f(-6.5,1);  glVertex2f(-6.5,0);  glEnd();  //E  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(-4,2);  glVertex2f(-2.5,2);  glVertex2f(-2.5,0);  glVertex2f(-4,0);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-3.5,1.55);  glVertex2f(-2.5,1.55);  glVertex2f(-2.5,1.25);  glVertex2f(-3.5,1.25);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-3.5,0.75);  glVertex2f(-2.5,0.75);  glVertex2f(-2.5,0.455);  glVertex2f(-3.5,0.45);  glEnd();  //P  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=1.5-.818;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x-0.81844,y+1.381 );  }  glEnd();  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(255,255,255);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=0.81844-0.56589;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x-0.81844,y+1.381 );  }  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(-1.5,2.5);  glVertex2f(0,2.5);  glVertex2f(0,2);  glVertex2f(-1.5,2);  glEnd();  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(-1.5,2);  glVertex2f(-1,2);  glVertex2f(-1,0);  glVertex2f(-1.5,0);  glEnd();  //T  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(1.5,2);  glVertex2f(1.5,0);  glVertex2f(2,0);  glVertex2f(2,2);  glEnd();  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(1,2);  glVertex2f(2.5,2);  glVertex2f(2.5,1.5);  glVertex2f(1,1.5);  glEnd();  //U  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(3.5,2);  glVertex2f(5.5,2);  glVertex2f(5.5,0);  glVertex2f(3.5,0);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(4,2);  glVertex2f(5,2);  glVertex2f(5,0.5);  glVertex2f(4,0.5);  glEnd();  //N  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(6.5,2);  glVertex2f(8.5,2);  glVertex2f(8.5,0);  glVertex2f(6.5,0);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(7,2);  glVertex2f(8,1);  glVertex2f(8,2);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(8,0);  glVertex2f(7,1);  glVertex2f(7,0);  glEnd();  //E  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(9.5,2);  glVertex2f(11,2);  glVertex2f(11,0);  glVertex2f(9.5,0);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(10,1.55);  glVertex2f(11,1.55);  glVertex2f(11,1.25);  glVertex2f(10,1.25);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(10,0.75);  glVertex2f(11,0.75);  glVertex2f(11,0.455);  glVertex2f(10,0.45);  glEnd();  }  void display()  {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  myName();  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");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  glutInitWindowSize(320, 320);// Set the window's initial width & height  glutDisplayFunc(display);// Register display callback handler for window re-paint  gluOrtho2D(-12,15,-4,7);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| Output: |

|  |
| --- |
| Text:BRAVO |
| Graph: |
| Code:  #include <windows.h> // for MS Windows  #include <GL/glut.h> // GLUT, include glu.h and gl.h  #include <math.h>  void myName()  {  //B  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=1.7;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+1.3,y+4 );  }  glEnd();  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(255,255,255);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=0.7;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+1.3,y+3.7);  }  glEnd();  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=1.7;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+1.3,y+1 );  }  glEnd();  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(255,255,255);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=0.7;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+1.3,y+1.3);  }  glEnd();  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(0.5,0);  glVertex2f(-1,0);  glVertex2f(-1,5);  glVertex2f(0.5,5);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(3,0);  glVertex2f(3,-1);  glVertex2f(-1,-1);  glVertex2f(-1,0);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(3,5);  glVertex2f(3,6);  glVertex2f(-1,6);  glVertex2f(-1,5);  glEnd();  //R  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=8.07-6.58;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+6.58,y+3.53 );  }  glEnd();  glBegin(GL\_POLYGON);  for(int i=0;i<200;i++)  {  glColor3ub(255,255,255);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=7.19-6.58;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+6.58,y+3.53 );  }  glEnd();  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(5,5);  glVertex2f(6,5);  glVertex2f(6,0);  glVertex2f(5,0);  glEnd();  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(6,2.6);  glVertex2f(8.6,0);  glVertex2f(7.5,0);  glVertex2f(6,1.5);  glEnd();  //A  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(11.5,5);  glVertex2f(12.5,5);  glVertex2f(14,0);  glVertex2f(13,0);  glVertex2f(12.7,1.4);  glVertex2f(11.3,1.4);  glVertex2f(11,0);  glVertex2f(10,0);  glEnd();  glColor3ub(255,255,255);  glBegin(GL\_POLYGON);  glVertex2f(12,4.2);  glVertex2f(12.6,2.2);  glVertex2f(11.4,2.2);  glEnd();  //V  glColor3ub(0,0,0);  glBegin(GL\_POLYGON);  glVertex2f(18,1.5);  glVertex2f(19,5);  glVertex2f(20,5);  glVertex2f(18.6,0);  glVertex2f(17.4,0);  glVertex2f(16,5);  glVertex2f(17,5);  glEnd();  //O  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(0,0,0);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=2.5;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+24,y+2.5);  }  glEnd();  glBegin(GL\_POLYGON);// Draw a Red 1x1 Square centered at origin  for(int i=0;i<200;i++)  {  glColor3ub(255,255,255);  float pi=3.1416;  float A=(i\*2\*pi)/200;  float r=1.5;  float x = r \* cos(A);  float y = r \* sin(A);  glVertex2f(x+24,y+2.5 );  }  glEnd();  }  void display()  {  glClearColor(1.0f, 1.0f, 1.0f, 1.0f); // Set background color to black and opaque  glClear(GL\_COLOR\_BUFFER\_BIT);  myName();  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");  //gluOrtho2D(-0.1,0.7,-0.1,0.3); // Create a window with the given title  glutInitWindowSize(320, 320);// Set the window's initial width & height  glutDisplayFunc(display);// Register display callback handler for window re-paint  gluOrtho2D(-14,32,-4,7);  glutMainLoop(); // Enter the event-processing loop  return 0;  } |
| Output: |