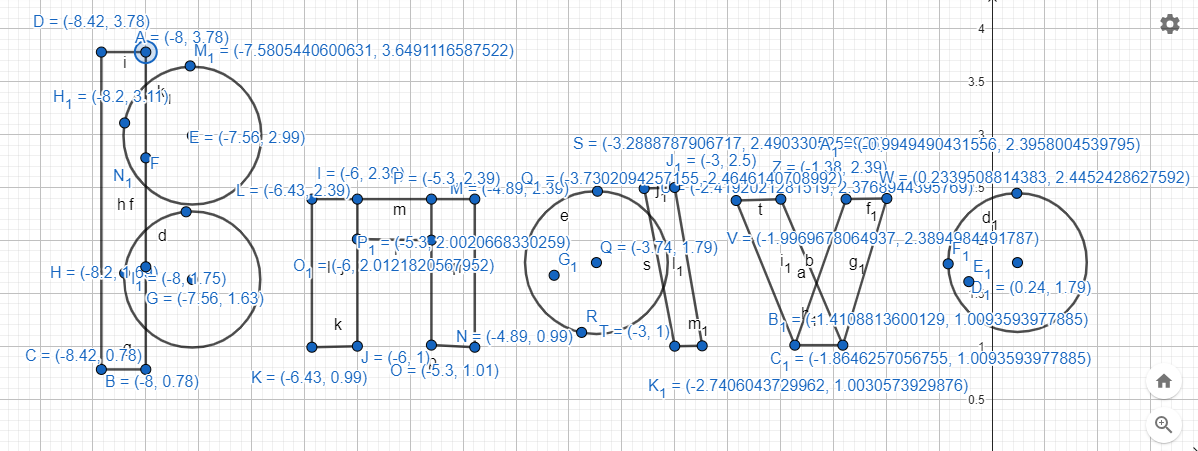
**BRAVO**

**Graph:**

****

**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Bravo()

{

//B

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-8.42,0.78);

glVertex2f(-8.42,3.78);

glVertex2f(-8,3.78);

glVertex2f(-8,0.78);

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=3.65-2.99;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x-7.56,y+2.99);

}

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=2.27-1.63;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x-7.56,y+1.63);

}

glEnd();

//r

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-6.43,0.99);

glVertex2f(-6.43,2.39);

glVertex2f(-6,2.39);

glVertex2f(-6,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-6,2.01);

glVertex2f(-6,2.39);

glVertex2f(-5.3,2.39);

glVertex2f(-5.3,2);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-5.3,1.01);

glVertex2f(-5.3,2.39);

glVertex2f(-4.89,2.39);

glVertex2f(-4.89,0.99);

glEnd();

//a

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-3,1);

glVertex2f(-3.3,2.5);

glVertex2f(-3,2.5);

glVertex2f(-2.74,1);

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=2.5-1.8;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x-3.74,y+1.8);

}

glEnd();

//v

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-1.9,1);

glVertex2f(-2.4,2.4);

glVertex2f(-1.9,2.4);

glVertex2f(-1.4,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-1.9,1);

glVertex2f(-1.4,2.4);

glVertex2f(-0.9,2.4);

glVertex2f(-1.4,1);

glEnd();

//o

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=2.5-1.79;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+0.24,y+1.79);

}

glEnd();

}

void display()

{

glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

Bravo();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL Bravo");

glutInitWindowSize(320,320);

glutDisplayFunc(display);

gluOrtho2D(-20,15,-20,15);

glutMainLoop();

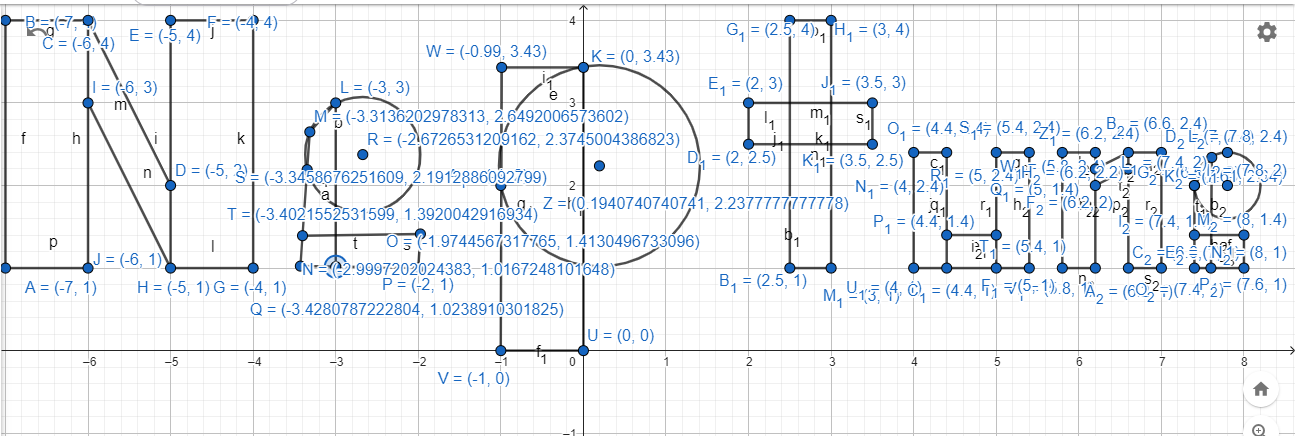
return 0;}

**Output:**

****

**NEPTUNE**

**Graph:**

****

**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Neptune()

{

//N

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-7,1);

glVertex2f(-7,4);

glVertex2f(-6,4);

glVertex2f(-6,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-5,1);

glVertex2f(-5,4);

glVertex2f(-4,4);

glVertex2f(-4,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-6,3);

glVertex2f(-6,4);

glVertex2f(-5,2);

glVertex2f(-5,1);

glEnd();

//e

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-3.43,1.02);

glVertex2f(-3.40,1.39);

glVertex2f(-2,1.41);

glVertex2f(-2,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-3.43,1.02);

glVertex2f(-3.31,2.65);

glVertex2f(-3,3);

glVertex2f(-3,1.02);

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=3-2.37;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x-2.67,y+2.37);

}

glEnd();

//p

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-1,0);

glVertex2f(-0.99,3.43);

glVertex2f(0,3.43);

glVertex2f(0,0);

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=3.43-2.24;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+0.19,y+2.24);

}

glEnd();

//t

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(2.5,1);

glVertex2f(2.5,4);

glVertex2f(3,4);

glVertex2f(3,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(2,2.5);

glVertex2f(2,3);

glVertex2f(3.5,3);

glVertex2f(3.5,2.5);

glEnd();

//u

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(4,1);

glVertex2f(4,2.4);

glVertex2f(4.4,2.4);

glVertex2f(4.4,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(4.4,1);

glVertex2f(4.4,1.4);

glVertex2f(5,1.4);

glVertex2f(5,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(5,1);

glVertex2f(5,2.4);

glVertex2f(5.4,2.4);

glVertex2f(5.4,1);

glEnd();

//n

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(5.8,1);

glVertex2f(5.8,2.4);

glVertex2f(6.2,2.4);

glVertex2f(6.2,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(6.2,2);

glVertex2f(6.2,2.2);

glVertex2f(6.6,2.4);

glVertex2f(6.6,2.2);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(6.6,1);

glVertex2f(6.6,2.4);

glVertex2f(7,2.4);

glVertex2f(7,1);

glEnd();

//e

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(7.4,1);

glVertex2f(7.4,1.4);

glVertex2f(8,1.4);

glVertex2f(8,1);

glEnd();

glColor3f(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(7.4,1);

glVertex2f(7.4,2);

glVertex2f(7.61,2.34);

glVertex2f(7.6,1);

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=2.4-2;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x+7.8,y+2);

}

glEnd();

}

void display()

{

glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

Neptune();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL Neptune");

glutInitWindowSize(320,320);

glutDisplayFunc(display);

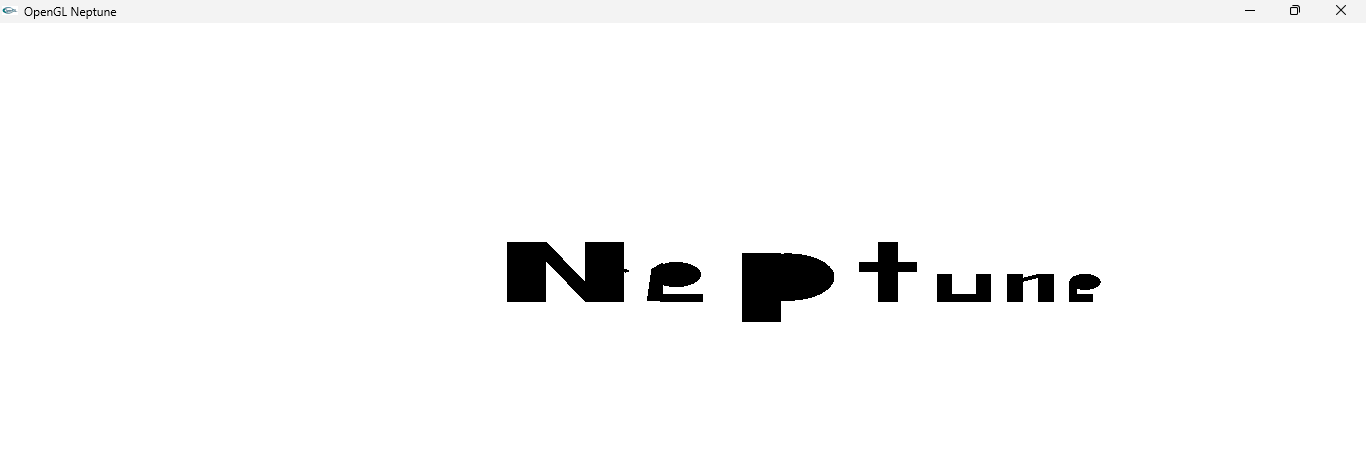
gluOrtho2D(-20,15,-20,15);

glutMainLoop();

return 0;

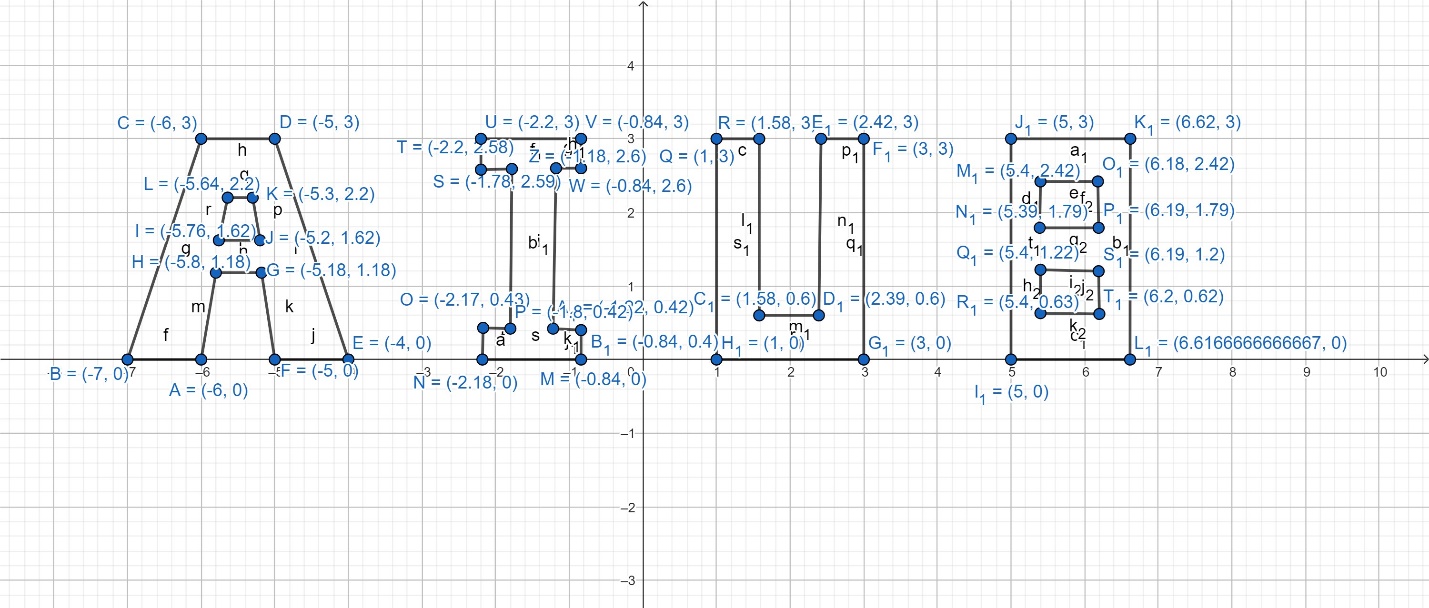
}

**Output:**

****

**AIUB**

**Graph:**



**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void AIUB()

{

glColor3ub(0,0,0);

glLineWidth(4);

glBegin(GL\_LINES);

//A

glVertex2f(-6,0);

glVertex2f(-7,0);

glVertex2f(-7,0);

glVertex2f(-6,3);

glVertex2f(-6,3);

glVertex2f(-5,3);

glVertex2f(-5,3);

glVertex2f(-4,0);

glVertex2f(-4,0);

glVertex2f(-5,0);

glVertex2f(-5,0);

glVertex2f(-5.18,1.18);

glVertex2f(-5.18,1.18);

glVertex2f(-5.8,1.18);

glVertex2f(-5.8,1.18);

glVertex2f(-6,0);

glVertex2f(-5.76,1.62);

glVertex2f(-5.2,1.62);

glVertex2f(-5.2,1.62);

glVertex2f(-5.3,2.2);

glVertex2f(-5.3,2.2);

glVertex2f(-5.64,2.2);

glVertex2f(-5.64,2.2);

glVertex2f(-5.76,1.62);

//I

glVertex2f(-0.84,0);

glVertex2f(-2.18,0);

glVertex2f(-2.18,0);

glVertex2f(-2.17,0.43);

glVertex2f(-2.17,0.43);

glVertex2f(-1.8,0.42);

glVertex2f(-1.8,0.42);

glVertex2f(-1.78,2.59);

glVertex2f(-1.78,2.59);

glVertex2f(-2.2,2.58);

glVertex2f(-2.2,2.58);

glVertex2f(-2.2,3);

glVertex2f(-2.2,3);

glVertex2f(-0.84,3);

glVertex2f(-0.84,3);

glVertex2f(-0.84,2.6);

glVertex2f(-0.84,2.6);

glVertex2f(-1.18,2.6);

glVertex2f(-1.18,2.6);

glVertex2f(-1.22,0.42);

glVertex2f(-1.22,0.42);

glVertex2f(-0.84,0.4);

glVertex2f(-0.84,0.4);

glVertex2f(-0.84,0);

//U

glVertex2f(1,3);

glVertex2f(1.58,3);

glVertex2f(1.58,3);

glVertex2f(1.58,0.6);

glVertex2f(1.58,0.6);

glVertex2f(2.39,0.6);

glVertex2f(2.39,0.6);

glVertex2f(2.42,3);

glVertex2f(2.42,3);

glVertex2f(3,3);

glVertex2f(3,3);

glVertex2f(3,0);

glVertex2f(3,0);

glVertex2f(1,0);

glVertex2f(1,0);

glVertex2f(1,3);

//B

glVertex2f(5,0);

glVertex2f(5,3);

glVertex2f(5,3);

glVertex2f(6.62,3);

glVertex2f(6.62,3);

glVertex2f(6.616,0);

glVertex2f(6.616,0);

glVertex2f(5,0);

glVertex2f(5.4,2.42);

glVertex2f(5.39,1.79);

glVertex2f(5.39,1.79);

glVertex2f(6.19,1.79);

glVertex2f(6.19,1.79);

glVertex2f(6.18,2.42);

glVertex2f(6.18,2.42);

glVertex2f(5.4,2.42);

glVertex2f(5.4,1.22);

glVertex2f(5.4,0.63);

glVertex2f(5.4,0.63);

glVertex2f(6.2,0.62);

glVertex2f(6.2,0.62);

glVertex2f(6.19,1.2);

glVertex2f(6.19,1.2);

glVertex2f(5.4,1.22);

glEnd();

}

void display()

{

glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

AIUB();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL AIUB");

glutInitWindowSize(320, 320);

glutDisplayFunc(display);

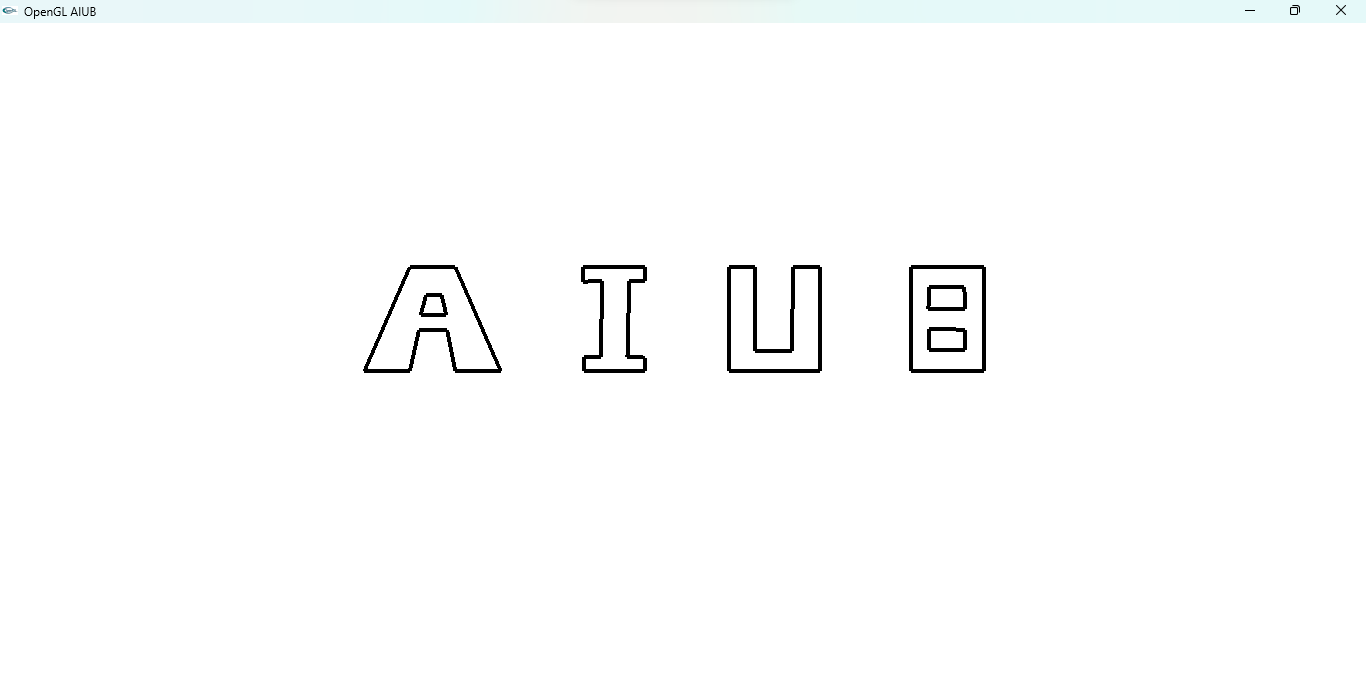
gluOrtho2D(-15,15,-10,10);

glutMainLoop();

return 0;

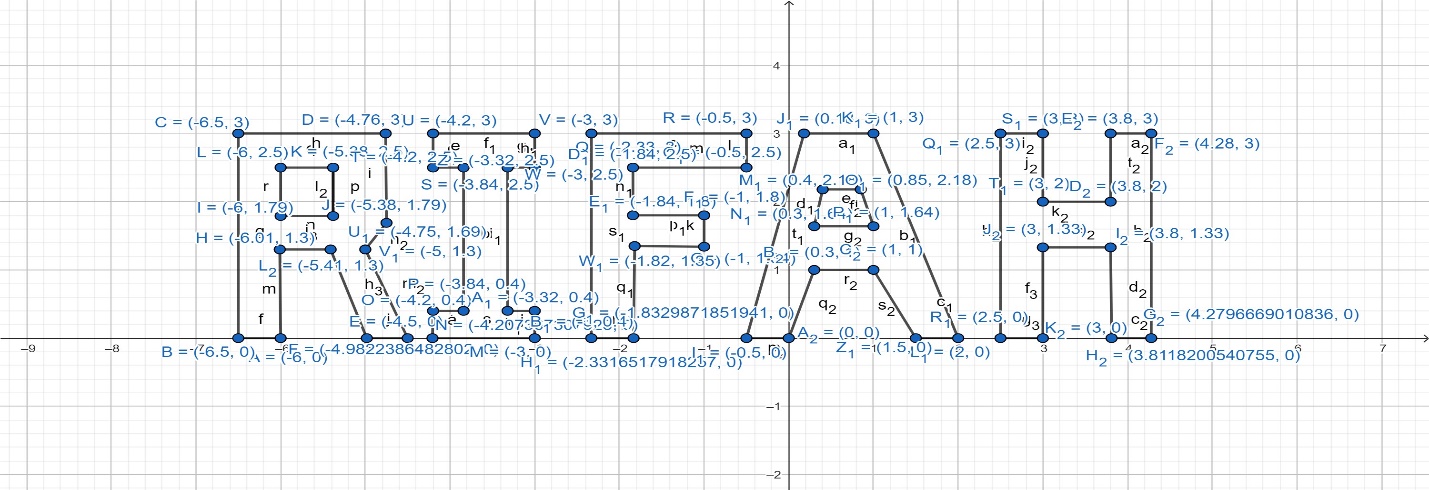
}

**Output:**



**YOUR NAME**

**Graph:**

****

**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void display()

{

glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

glLineWidth(4);

glBegin(GL\_LINES);

glColor3f(1.0f, 0.0f, 0.0f);

//R

glVertex2f(-6,0);

glVertex2f(-6.5,0);

glVertex2f(-6.5,0);

glVertex2f(-6.5,3);

glVertex2f(-6.5,3);

glVertex2f(-4.76,3);

glVertex2f(-4.76,3);

glVertex2f(-4.75,1.69);

glVertex2f(-4.75,1.69);

glVertex2f(-5,1.3);

glVertex2f(-5,1.3);

glVertex2f(-4.5,0);

glVertex2f(-4.5,0);

glVertex2f(-4.98,0);

glVertex2f(-4.98,0);

glVertex2f(-5.41,1.3);

glVertex2f(-5.41,1.3);

glVertex2f(-6.01,1.3);

glVertex2f(-6.01,1.3);

glVertex2f(-6,0);

glVertex2f(-6,1.79);

glVertex2f(-6,2.5);

glVertex2f(-6,2.5);

glVertex2f(-5.38,2.5);

glVertex2f(-5.38,2.5);

glVertex2f(-5.38,1.79);

glVertex2f(-5.38,1.79);

glVertex2f(-6,1.79);

glVertex2f(-6,1.79);

glVertex2f(-6,2.5);

//I

glVertex2f(-3,0);

glVertex2f(-4.207,0);

glVertex2f(-4.207,0);

glVertex2f(-4.2,0.4);

glVertex2f(-4.2,0.4);

glVertex2f(-3.84,0.4);

glVertex2f(-3.84,0.4);

glVertex2f(-3.84,2.5);

glVertex2f(-3.84,2.5);

glVertex2f(-4.2,2.5);

glVertex2f(-4.2,2.5);

glVertex2f(-4.2,3);

glVertex2f(-4.2,3);

glVertex2f(-3,3);

glVertex2f(-3,3);

glVertex2f(-3,2.5);

glVertex2f(-3,2.5);

glVertex2f(-3.32,2.5);

glVertex2f(-3.32,2.5);

glVertex2f(-3.32,0.4);

glVertex2f(-3.32,0.4);

glVertex2f(-3,0.4);

glVertex2f(-3,0.4);

glVertex2f(-3,0);

//F

glVertex2f(-1.83,0);

glVertex2f(-2.33,0);

glVertex2f(-2.33,0);

glVertex2f(-2.33,3);

glVertex2f(-2.33,3);

glVertex2f(-0.5,3);

glVertex2f(-0.5,3);

glVertex2f(-0.5,2.5);

glVertex2f(-0.5,2.5);

glVertex2f(-1.84,2.5);

glVertex2f(-1.84,2.5);

glVertex2f(-1.84,1.8);

glVertex2f(-1.84,1.8);

glVertex2f(-1,1.8);

glVertex2f(-1,1.8);

glVertex2f(-1,1.34);

glVertex2f(-1,1.34);

glVertex2f(-1.82,1.35);

glVertex2f(-1.82,1.35);

glVertex2f(-1.83,0);

//A

glVertex2f(0,0);

glVertex2f(-0.5,0);

glVertex2f(-0.5,0);

glVertex2f(-0.18,3);

glVertex2f(-0.18,3);

glVertex2f(1,3);

glVertex2f(1,3);

glVertex2f(2,0);

glVertex2f(2,0);

glVertex2f(1.5,0);

glVertex2f(1.5,0);

glVertex2f(1,1);

glVertex2f(1,1);

glVertex2f(0.3,1);

glVertex2f(0.3,1);

glVertex2f(0,0);

glVertex2f(0.3,1.64);

glVertex2f(0.4,2.18);

glVertex2f(0.4,2.18);

glVertex2f(0.85,2.18);

glVertex2f(0.85,2.18);

glVertex2f(1,1.64);

glVertex2f(1,1.64);

glVertex2f(0.3,1.64);

//H

glVertex2f(3,0);

glVertex2f(2.5,0);

glVertex2f(2.5,0);

glVertex2f(2.5,3);

glVertex2f(2.5,3);

glVertex2f(3,3);

glVertex2f(3,3);

glVertex2f(3,2);

glVertex2f(3,2);

glVertex2f(3.8,2);

glVertex2f(3.8,2);

glVertex2f(3.8,3);

glVertex2f(3.8,3);

glVertex2f(4.28,3);

glVertex2f(4.28,3);

glVertex2f(4.279,0);

glVertex2f(4.279,0);

glVertex2f(3.81,0);

glVertex2f(3.81,0);

glVertex2f(3.8,1.33);

glVertex2f(3.8,1.33);

glVertex2f(3,1.33);

glVertex2f(3,1.33);

glVertex2f(3,0);

glEnd();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL YOUR NAME");

glutInitWindowSize(320, 320);

glutDisplayFunc(display);

gluOrtho2D(-15,15,-10,10);

glutMainLoop();

return 0;

}

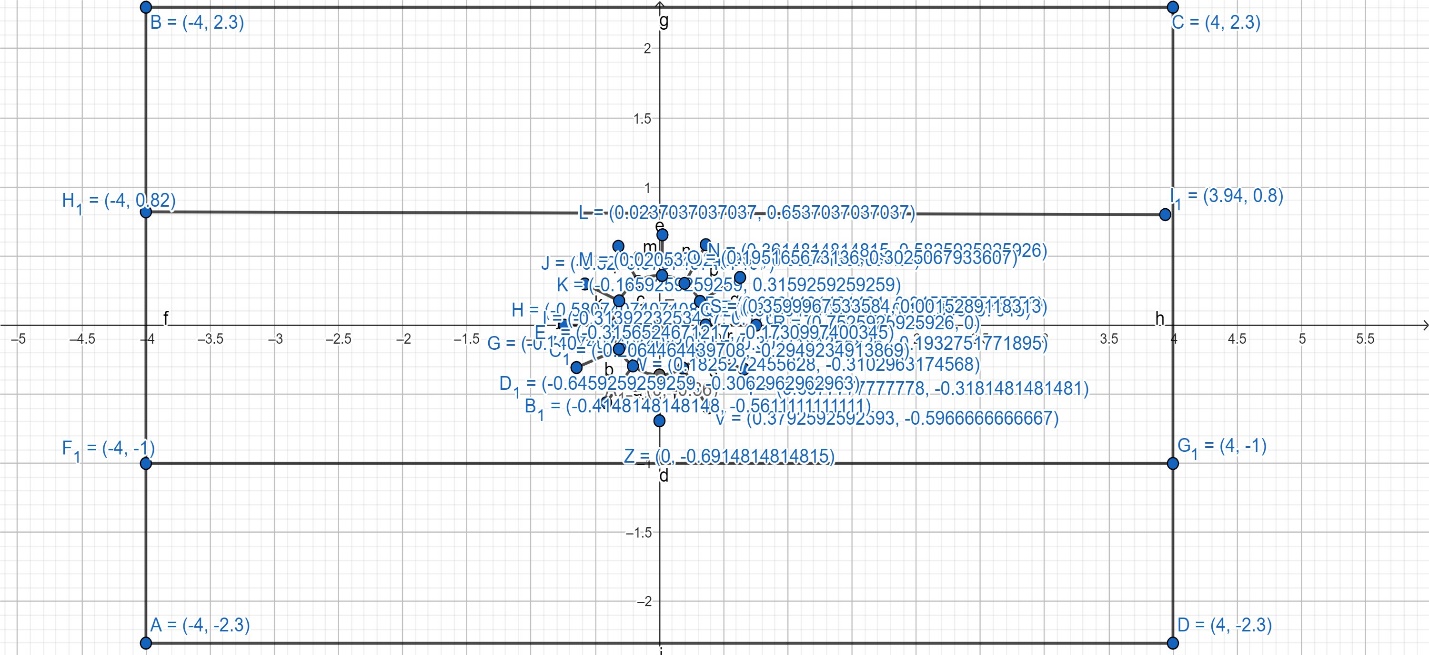
**Output:**

**A screenshot of a computer

Description automatically generated**

**Flag 1**

**Graph:**



**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Rectangle\_Circle()

{

glBegin(GL\_POLYGON);

glColor3f(0.38,0.81,0.85);

glVertex2f(-4,-2.3);

glVertex2f(-4,-0.78);

glVertex2f(4,-0.78);

glVertex2f(4,-2.3);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(1,1,1);

glVertex2f(-1,0);

glVertex2f(-0.82,0.48);

glVertex2f(4,0.78);

glVertex2f(4,-0.78);

glColor3ub(246, 220, 4);

glLineWidth(3.5);

glBegin(GL\_LINES);

glVertex2f(-0.36,0);

glVertex2f(-0.74,0);

glVertex2f(-0.31,0.176);

glVertex2f(-0.58,0.298);

glVertex2f(-0.1659,0.3159);

glVertex2f(-0.32,0.57);

glVertex2f(0.02,0.3594);

glVertex2f(0.02,0.65);

glVertex2f(0.195,0.30);

glVertex2f(0.36,0.58);

glVertex2f(0.316,0.17);

glVertex2f(0.628,0.345);

glVertex2f(0.359,0.001);

glVertex2f(0.75,0);

glVertex2f(0.30,-0.19);

glVertex2f(0.657,-0.318);

glVertex2f(0.18,-0.31);

glVertex2f(0.379,-0.596);

glVertex2f(0,-0.36);

glVertex2f(0,-0.69);

glVertex2f(-0.206,-0.294);

glVertex2f(-0.41,-0.56);

glVertex2f(-0.315,-0.173);

glVertex2f(-0.645,-0.306);

glEnd();

glBegin(GL\_POLYGON);

glColor3f(1,1,0);

for(int i=0;i<200;i++)

{

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=0.36;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

glEnd();

glBegin(GL\_POLYGON);

glColor3f(0.38,0.81,0.85);

glVertex2f(-4,0.78);

glVertex2f(-4,2.3);

glVertex2f(4,2.3);

glVertex2f(4,0.78);

glEnd();

}

void display()

{

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(1.0, 1.0, 1.0, 1.0);

glClear(GL\_COLOR\_BUFFER\_BIT);

Rectangle\_Circle();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL FLAG");

//gluOrtho2D(-0.1,0.7,-0.1,0.3);

glutInitWindowSize(320, 320);

glutDisplayFunc(display);

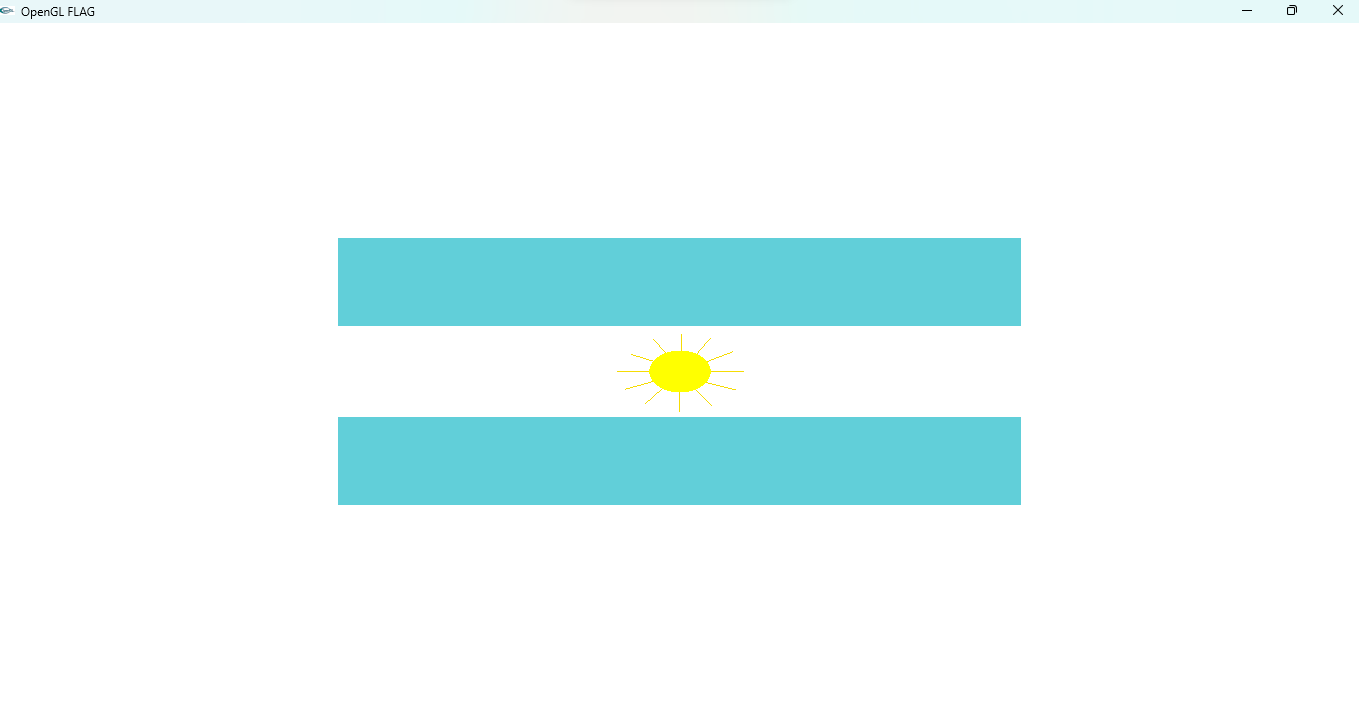
gluOrtho2D(-8,8,-6,6);

glutMainLoop();

return 0;

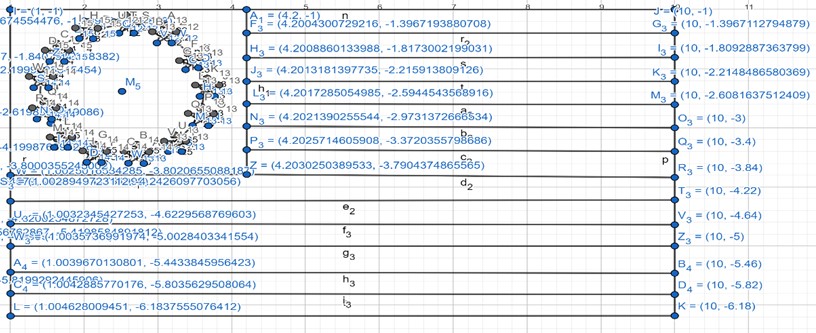
}

**Output:**



**Flag 2**

**Graph:**



**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void flag\_2()

{

//4th

//border

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\_2();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("Flag\_2");

glutInitWindowSize(320,320);

glutDisplayFunc(display);

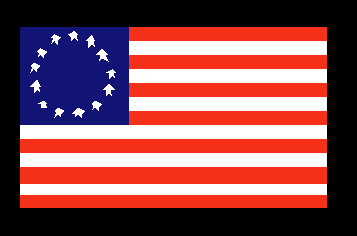
gluOrtho2D(-20,20,-10,10);

glutMainLoop();

return 0;

}

**Output:**

****

**Flag 3**

**Graph:**

A diagram of a square with a square and a square with a square and a square with a square and a square with a square and a square with a square and a square with a square and

Description automatically generated

**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void flag\_3()

{

//1st

//border

glColor3ub(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(2,5);

glVertex2f(10,5);

glVertex2f(10,1);

glVertex2f(2,1);

glEnd();

//redfull

glColor3ub(244,16,8);

glBegin(GL\_POLYGON);

glVertex2f(2,3.5);

glVertex2f(5.13,3.5);

glVertex2f(5.13,5);

glVertex2f(10,5);

glVertex2f(10,1);

glVertex2f(2,1);

glEnd();

//graymiddle

glColor3ub(228,226,225);

glBegin(GL\_POLYGON);

glVertex2f(2,3.5);

glVertex2f(2,3.63);

glVertex2f(3,3.63);

glVertex2f(4,3.63);

glVertex2f(5,3.63);

glVertex2f(5,5);

glVertex2f(5.13,5);

glVertex2f(5.13,3.5);

glEnd();

//blue

glColor3ub(29,59,141);

glBegin(GL\_POLYGON);

glVertex2f(2,3.63);

glVertex2f(2,5);

glVertex2f(3,5);

glVertex2f(3,3.63);

glEnd();

//grayup

glColor3ub(228,226,225);

glBegin(GL\_POLYGON);

glVertex2f(3,3.63);

glVertex2f(3,5);

glVertex2f(4,5);

glVertex2f(4,3.63);

glEnd();

//red

glColor3ub(244,16,8);

glBegin(GL\_POLYGON);

glVertex2f(4,3.63);

glVertex2f(4,5);

glVertex2f(5,5);

glVertex2f(5,3.63);

glEnd();

//gray1up

glColor3ub(228,226,225);

glBegin(GL\_POLYGON);

glVertex2f(7.5,3.5);

glVertex2f(8.5,3.5);

glVertex2f(8,2.95);

glEnd();

//gray2right

glColor3ub(228,226,225);

glBegin(GL\_POLYGON);

glVertex2f(8.5,3.3);

glVertex2f(8.5,2.5);

glVertex2f(8.13,2.86);

glEnd();

//gray3down

glColor3ub(228,226,225);

glBegin(GL\_POLYGON);

glVertex2f(8,2.75);

glVertex2f(8.5,2.38);

glVertex2f(7.5,2.38);

glEnd();

//gray4left

glColor3ub(228,226,225);

glBegin(GL\_POLYGON);

glVertex2f(7.5,3.3);

glVertex2f(7.89,2.85);

glVertex2f(7.5,2.5);

glEnd();

}

void display()

{

glClearColor(1,1,1,1);

glClear(GL\_COLOR\_BUFFER\_BIT);

flag\_3();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("Flag\_3");

glutInitWindowSize(320, 320);

glutDisplayFunc(display);

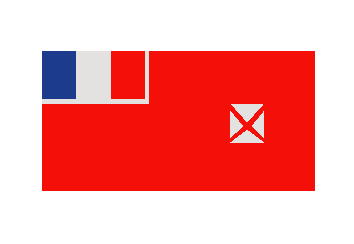
gluOrtho2D(-20,20,-10,10);

glutMainLoop();

return 0;

}

**Output:**

****

**Flag 4**

**Graph:**

A graph paper with a square and a circle

Description automatically generated

**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void Rectangle()

{

glColor3ub(11, 121, 26);

glBegin(GL\_POLYGON);

glVertex2f(-4.5,-2.3);

glVertex2f(-4.5,2.3);

glVertex2f(4.5,2.3);

glVertex2f(4.5,-2.3);

glEnd();

}

void Rectangle2()

{

glColor3ub(255, 255, 0);

glBegin(GL\_POLYGON);

glVertex2f(-2.5,0);

glVertex2f(0,1.8);

glVertex2f(2.5,0);

glVertex2f(0,-1.8);

glEnd();

}

void Circle()

{

glBegin(GL\_POLYGON);

for(int i=0;i<200;i++)

{

glColor3ub(34, 13, 120);

float pi=3.1416;

float A=(i\*2\*pi)/200;

float r=1.194;

float x = r \* cos(A);

float y = r \* sin(A);

glVertex2f(x,y );

}

glEnd();

}

void display()

{

glClearColor(1.0f, 1.0f, 1.0f, 1.0f);

glClear(GL\_COLOR\_BUFFER\_BIT);

Rectangle();

Rectangle2();

Circle();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("OpenGL FLAG");

//gluOrtho2D(-0.1,0.7,-0.1,0.3);

glutInitWindowSize(320, 320);

glutDisplayFunc(display);

gluOrtho2D(-8,8,-6,6);

glutMainLoop();

return 0;

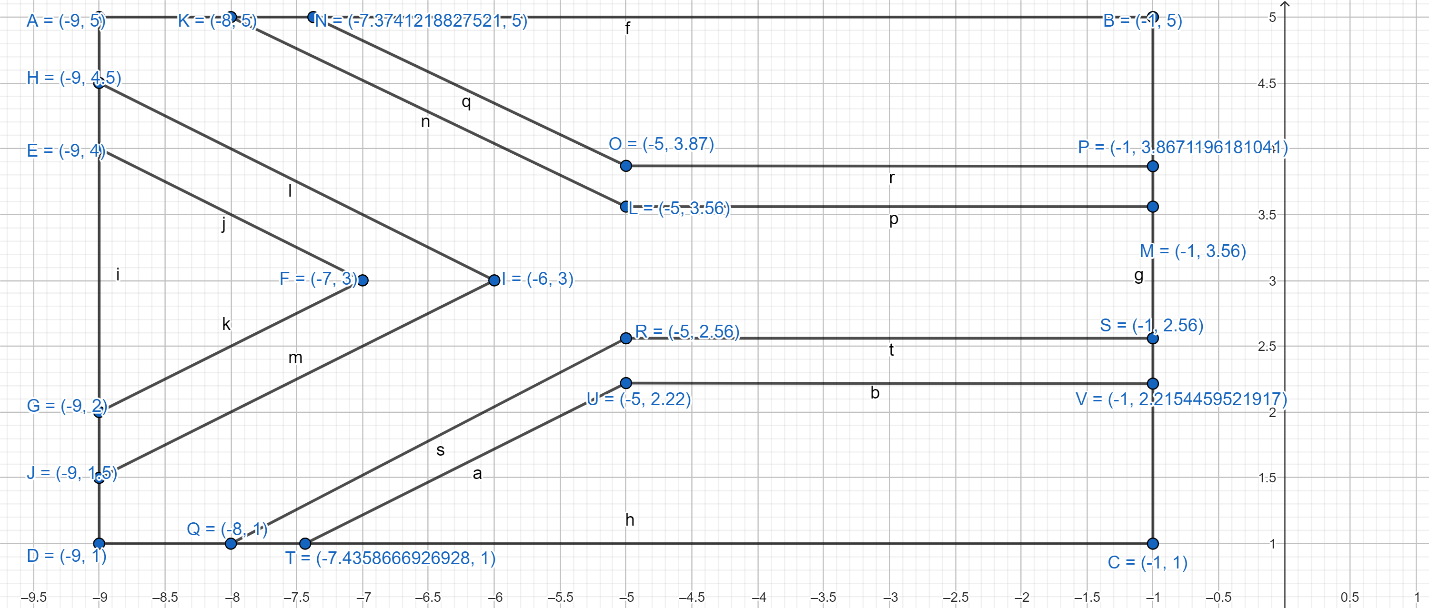
}

**Output:**

****

**Flag 5**

**Graph:**



**Code:**

#include <windows.h>

#include <GL/glut.h>

#include <math.h>

void flag\_5()

{

//2nd

//border

glColor3ub(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-9,5);

glVertex2f(-1,5);

glVertex2f(-1,1);

glVertex2f(-9,1);

glEnd();

//black

glColor3ub(0,0,0);

glBegin(GL\_POLYGON);

glVertex2f(-9,2);

glVertex2f(-9,4);

glVertex2f(-7,3);

glEnd();

//yellow

glColor3ub(255,195,0);

glBegin(GL\_POLYGON);

glVertex2f(-7,3);

glVertex2f(-9,4);

glVertex2f(-9,4.5);

glVertex2f(-6,3);

glVertex2f(-9,1.5);

glVertex2f(-9,2);

glEnd();

//green

glColor3ub(56,119,34);

glBegin(GL\_POLYGON);

glVertex2f(-6,3);

glVertex2f(-9,4.5);

glVertex2f(-9,5);

glVertex2f(-8,5);

glVertex2f(-5,3.56);

glVertex2f(-1,3.56);

glVertex2f(-1,2.56);

glVertex2f(-5,2.56);

glVertex2f(-8,1);

glVertex2f(-9,1);

glVertex2f(-9,1.5);

glEnd();

//whiteup

glColor3ub(255,255,255);

glBegin(GL\_POLYGON);

glVertex2f(-8,5);

glVertex2f(-7.374,5);

glVertex2f(-5,3.87);

glVertex2f(-1,3.87);

glVertex2f(-1,3.56);

glVertex2f(-5,3.56);

glEnd();

//whitedown

glColor3ub(255,255,255);

glBegin(GL\_POLYGON);

glVertex2f(-8,1);

glVertex2f(-5,2.56);

glVertex2f(-1,2.56);

glVertex2f(-1,2.22);

glVertex2f(-5,2.22);

glVertex2f(-7.44,1);

glEnd();

//red

glColor3ub(244,16,8);

glBegin(GL\_POLYGON);

glVertex2f(-7.374,5);

glVertex2f(-1,5);

glVertex2f(-1,3.87);

glVertex2f(-5,3.87);

glEnd();

//blue

glColor3ub(29,59,141);

glBegin(GL\_POLYGON);

glVertex2f(-7.44,1);

glVertex2f(-5,2.22);

glVertex2f(-1,2.22);

glVertex2f(-1,1);

glEnd();

void display()

{

glClearColor(1,1,1,1);

glClear(GL\_COLOR\_BUFFER\_BIT);

flag\_5();

glFlush();

}

int main(int argc, char\*\* argv)

{

glutInit(&argc, argv);

glutCreateWindow("Flag\_5");

glutInitWindowSize(320,320);

glutDisplayFunc(display);

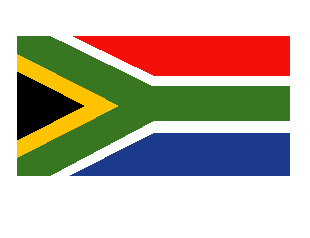
gluOrtho2D(-20,20,-10,10);

glutMainLoop();

return 0;

}

**Output:**

****