NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. VIth Sem

Subject Code: CS-317

Subject Name: Graphics and Multimedia Lab

Submitted By:

Name	: Subhojit Ghimire
Sch. Id.	: 1912160
Branch	: CSE – B

1. **Implementation of Midpoint ellipse drawing algorithm**
    a. **Divide the coordinate axes into four quadrants and then draw only a portion of an ellipse with the major axis as X-axis.**
    b. **Draw another portion of the ellipse with the major axis as Y-axis to the quadrant diagonally opposite to each other.**

Assign different colours to them respectively.

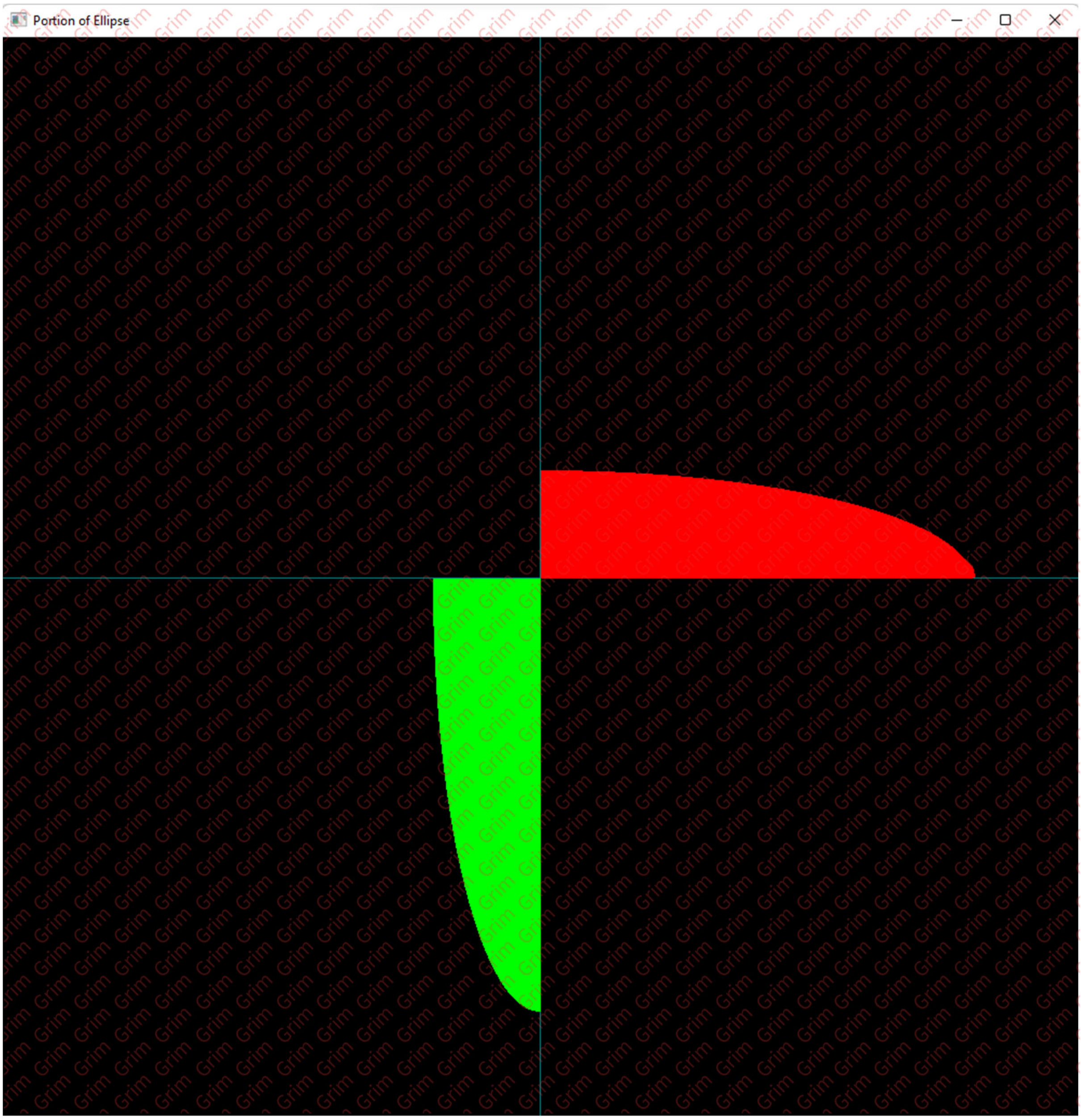
➔ **CODE:**

```
#include <GL/glut.h>
#include <stdio.h>

GLint offsetX, offsetY, centerX, centerY;
GLint w, x, y, z;

void myinit (void) {
      glClear (GL_COLOR_BUFFER_BIT);
      glClearColor (0.0, 0.0, 0.0, 0.0);
      glMatrixMode (GL_PROJECTION);
      glLoadIdentity ();
      gluOrtho2D (-500, 500, -500, 500);
      glColor3f (0.0f, 0.5f, 0.5f);

      glBegin (GL_LINES);
      glVertex3f (-500.0, 0.0, 0.0);
      glVertex3f (500.0, 0.0, 0.0);
    glEnd ();

    glBegin (GL_LINES);
      glVertex3f (0.0, -500.0, 0.0);
      glVertex3f (0.0, 500.0, 0.0);
    glEnd ();
}

void setPixel (GLint X, GLint Y) {
      if (w && (X >= 0 && Y >= 0))
            glVertex2i (X, Y);
    if (x && (X <= 0 && Y >= 0))
      glVertex2i (X, Y);
    if (y && (X <= 0 && Y <= 0))
      glVertex2i (X, Y);
    if (z && (X >= 0 && Y <= 0))
      glVertex2i (X, Y);
}

void setAxesPixel () {
      if (w) {
            glVertex2i (offsetX, 0);
          glVertex2i (0, 0);
          glVertex2i (0, offsetY);
      }
    if (x) {
      glVertex2i (-offsetX, 0);
          glVertex2i (0, 0);
          glVertex2i (0, offsetY);
      }
    if (y) {
      glVertex2i (-offsetX, 0);
          glVertex2i (0, 0);
          glVertex2i (0, -offsetY);
      }
    if (z) {
      glVertex2i (offsetX, 0);
          glVertex2i (0, 0);
          glVertex2i (0, -offsetY);
      }
}

void ellipseMidPoint (GLint W, GLint X, GLint Y, GLint Z) {
      w = W;      x = X;      y = Y;      z = Z;

      glBegin (GL_POLYGON);
          float Xk = 0;
          float Yk = offsetY;
          float p1 = offsetY * offsetY - (offsetX * offsetX) * offsetY
+ (offsetX * offsetX) * (0.25);
          //slope
          float dx = 2 * (offsetY * offsetY) * Xk;
          float dy = 2 * (offsetX * offsetX) * Yk;
          float p2 = (offsetY * offsetY) * (Xk + 0.5) * (Xk + 0.5) +
(offsetX * offsetX) * (Yk - 1) * (Yk - 1) - (offsetX * offsetX ) *
(offsetY * offsetY);

            while (dx < dy) {
              setPixel (centerX + Xk,centerY + Yk);
              setPixel (centerX - Xk,centerY + Yk);
              setPixel (centerX + Xk,centerY - Yk);
              setPixel (centerX - Xk,centerY - Yk);
              if (p1 < 0) {
                  Xk = Xk + 1;
                  dx = 2 * (offsetY * offsetY) * Xk;
                  p1 = p1 + dx + (offsetY * offsetY);
              }
              else {
                  Xk = Xk + 1;
```

```
         Yk = Yk - 1;
         dx =  2 * (offsetY * offsetY) * Xk;
         dy =  2 * (offsetX * offsetX) * Yk;
         p1 = p1 + dx - dy + (offsetY * offsetY);
       }
     }
     while (Yk > 0) {
       setPixel (centerX + Xk,centerY + Yk);
       setPixel (centerX - Xk,centerY + Yk);
       setPixel (centerX + Xk,centerY - Yk);
       setPixel (centerX - Xk,centerY - Yk);
       if (p2 > 0) {
         Xk = Xk;
         Yk = Yk - 1;
         dy = 2 * (offsetX * offsetX) * Yk;
         p2 = p2 - dy + (offsetX * offsetX);
       }
       else {
         Xk = Xk + 1;
         Yk = Yk - 1;
         dy = dy - 2 * (offsetX * offsetX) ;
         dx = dx + 2 * (offsetY * offsetY) ;
         p2 = p2 + dx -
         dy + (offsetX * offsetX);
       }
     }
      setAxesPixel ();
   glEnd ();
}

void display () {
   centerX = 0;centerY = 0; offsetX = 400; offsetY = 100;
  glColor3f (1.0, 0.0, 0.0);
  ellipseMidPoint (1, 0, 0, 0);

   centerX = 0;centerY = 0; offsetX = 100; offsetY = 400;
  glColor3f (0.0, 1.0, 0.0);
  ellipseMidPoint (0, 0, 1, 0);

  glFlush ();
}

int main (int argc, char **argv) {
   glutInit (&argc, argv);
  glutInitWindowSize (1000, 1000);
  glutInitWindowPosition (500, 0);
  glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
  glutCreateWindow ("Portion of Ellipse");
```

```
      myinit ();
    glutDisplayFunc (display);
    glutMainLoop ();
    return 0;
}
```

OUTPUT:


Portion of Ellipse