CSO351: Computer Graphics

Lab Assignment 2.Ellipse.(d):
Ellispe Generation using Midpoint Algorithm

Objective:

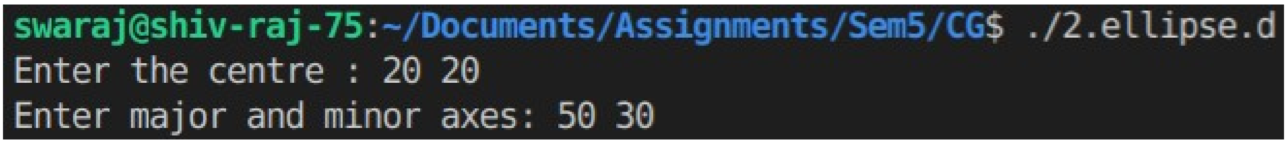
Write a program in C/C++ for implementation of ellipse generation using midpoint algorithm.

Algorithm:

- **Step 1:** Input center of ellipse (h, k) and major axis a and minor axis b.
- **Step 2:** Initially, we assume ellipse to be centered at origin and the first point as: $(x_0, y_0) = (0, b)$
- **Step 3:** Obtain the initial decision parameter for region 1 as:
  $p1_0 = b^2 + (1/4)a^2 - a^2b^2$
- **Step 4:** For every x_k position in region 1:
```
if ($p1_k < 0$)
    $p1_{k+1} = p1_k + 2b^2x_{k+1} + b^2$;
else
    $p1_{k+1} = p1_k + 2b^2x_{k+1} - 2a^2y_{k+1} + b^2$;
```
- **Step 5:** Obtain the initial value in region 2 using the last point (x_0 , y_0) of region 1 as: $p2_0 = b^2(x_0+1/2)^2+a^2(y_0-1)^2 - a^2b^2$
- **Step 6:** At each y_k in region 2 starting at k = 0 perform the following task:
```
if ($p2_k > 0$)
    $p2_{k+1} = p2_k - 2b^2y_{k+1} + a^2$;
else
    $p2_{k+1} = p2_k + 2b^2x_{k+1} - 2a^2y_{k+1} + a^2$;
```
- **Step 7:** Now obtain the symmetric points in the three quadrants and plot the coordinate value as: `x = x + h; y = y + k;`
- **Step 8:** Repeat the steps for region 1 until $2b^2x >= 2a^2y$.

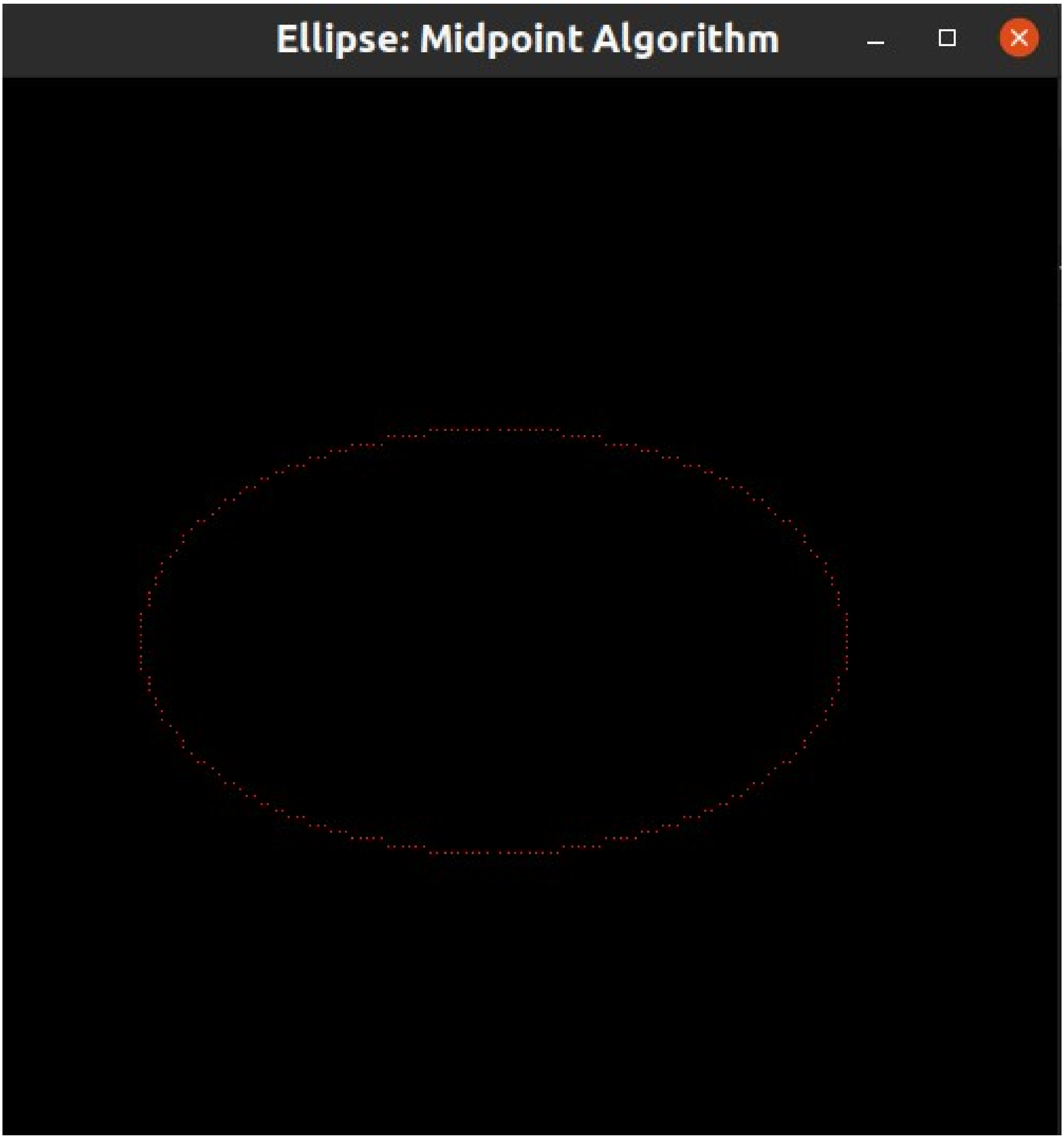
Result:

Input:



```
swaraj@shiv-raj-75:~/Documents/Assignments/Sem5/CG$ ./2.ellipse.d
Enter the centre : 20 20
Enter major and minor axes: 50 30
```

Output:



Conclusion:

- The mid-point ellipse algorithm has simple and easy implementation as it only includes addition operations.
- It is time-consuming.

Appendix: Code

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

using namespace std;
int h, k, a, b;

void init(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(-50, 100, -50, 100);
}

void plot(int x, int y)
{
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}

void ellipse(int h, int k, int a, int b)
{
    int x, y;
    float p;
    p = b * b - a * a * b + a * a / 4;
    x = 0;
    y = b;
    while (2.0 * b * b * x <= 2.0 * a * a * y)
    {
        if (p < 0)
        {
            x++;
            p = p + 2 * b * b * x + b * b;
        }
        else
        {
            x++;
            y--;
            p = p + 2 * b * b * x - 2 * a * a * y - b * b;
        }
        plot(h + x, k + y);
        plot(h + x, k - y);
        plot(h - x, k + y);
        plot(h - x, k - y);
    }
    p = b * b * (x + 0.5) * (x + 0.5) + a * a * (y - 1) * (y - 1)
        - a * a * b * b;
    while (y > 0)
    {
        if (p <= 0)
        {
            x++;
            y--;
            p = p + 2 * b * b * x - 2 * a * a * y + a * a;
        }
        else
        {
            y--;
            p = p - 2 * a * a * y + a * a;
        }
        plot(h + x, k + y);
        plot(h + x, k - y);
        plot(h - x, k + y);
        plot(h - x, k - y);
    }
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 0.0, 0.0);
    glPointSize(1.0);
    ellipse(h, k, a, b);
    glFlush();
}

int main(int argc, char **argv)
{
    cout << "Enter the centre : ";
    cin >> h >> k;
    cout << "Enter major and minor axes: ";
    cin >> a >> b;
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Ellipse: Midpoint Algorithm");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
}
```