Lab Assignment-1
Submitted By:
Name: Himesh Maniyar
ID: 2020UCP1776

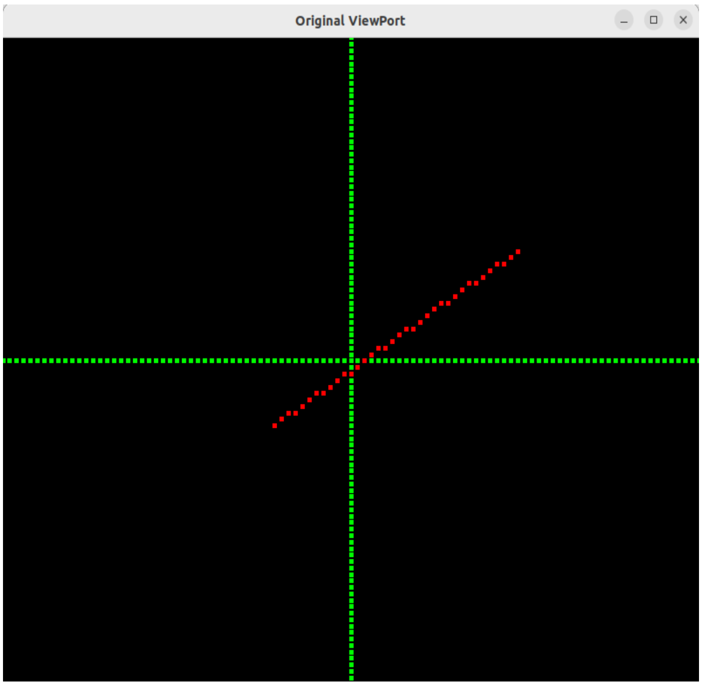
Program-1

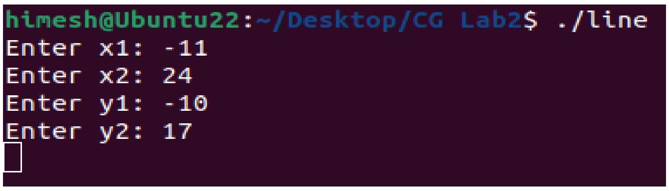
Code:

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

int X1,X2,Y1,Y2;

void myInit(void)
{
   glClearColor(0.0, 0.0, 0.0, 1.0);
   glColor3f(0.0, 0.0, 1.0);

   glPointSize(5.0);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();

   gluOrtho2D(-50, 50, -50, 50);
}

void display(void)
{
   glClear(GL_COLOR_BUFFER_BIT);
   glBegin(GL_POINTS);
   glColor3f(0.0,1.0,0.0);
   // PLotting xy plane
   for (int y = -400; y <= 400; y += 1){
         glVertex2f(0, y);
   }
   for (int x = -300; x <= 300; x += 1){
         glVertex2f(x, 0);
   }
   glEnd();

   int di = 2*(Y2-Y1)-(X2-X1);
   int y=Y1;
   glBegin(GL_POINTS);
   glColor3f(1,0,0);
   for(int x = X1;x<=X2&&y<=Y2;x++){
      glVertex2f(x,y);

      if(di>0){
         y++;
         di += 2*((Y2-Y1)-(X2-X1));
      }
      else{
         di+= 2*(Y2-Y1);
```

```
        }
    }
    glEnd();
    glFlush();
}
int main(int argc, char **argv)
{
    printf("Enter x1: ");
    scanf("%d",&X1);
    printf("Enter x2: ");
    scanf("%d",&X2);
    printf("Enter y1: ");
    scanf("%d",&Y1);
    printf("Enter y2: ");
    scanf("%d",&Y2);

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(1000, 1000);
    glutInitWindowPosition(0, 0);

    glutCreateWindow("Original ViewPort");
    myInit();

    glutDisplayFunc(display);
    glutMainLoop();
}
```

Output:





```
himesh@Ubuntu22:~/Desktop/CG Lab2$ ./line
Enter x1: -11
Enter x2: 24
Enter y1: -10
Enter y2: 17
```

Lab Assignment-1
Submitted By:
Name: Himesh Maniyar
ID: 2020UCP1776

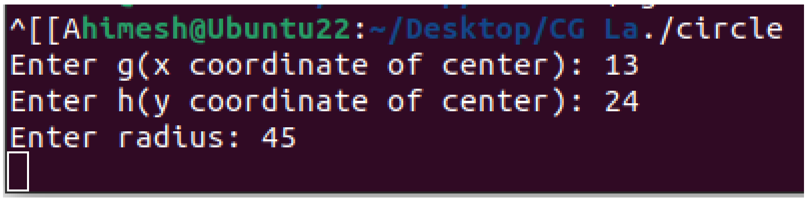
Program-2

Code:

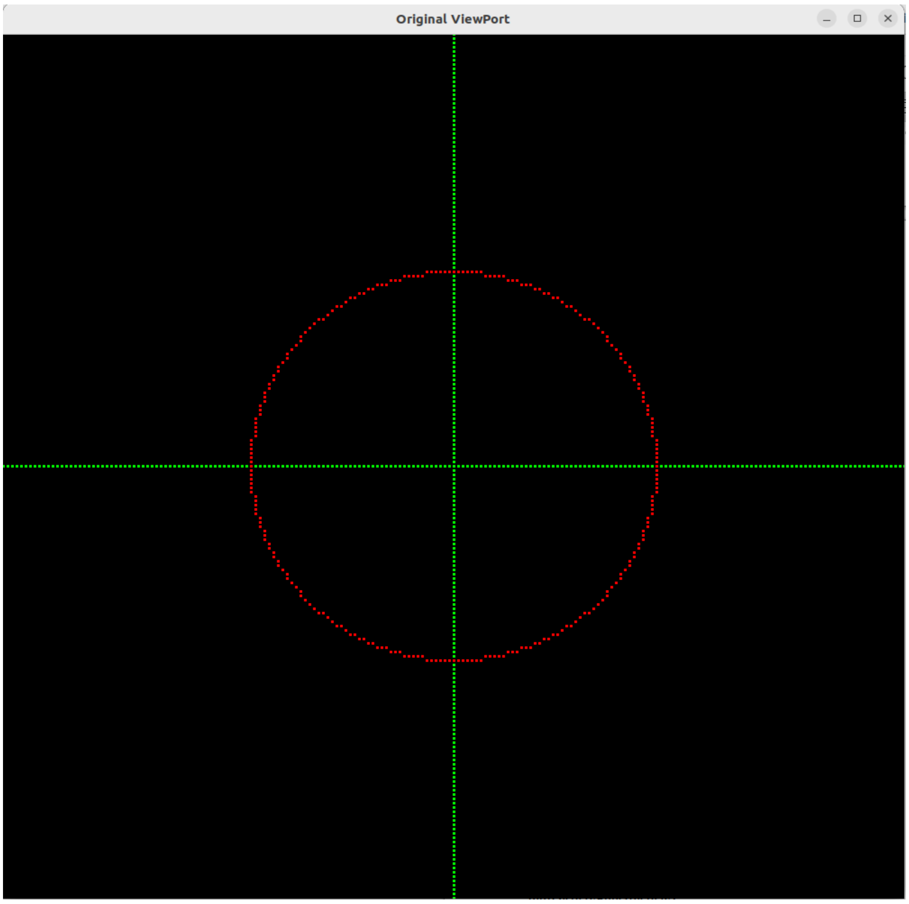
```
#include <stdio.h>
#include <GL/glut.h>

int g,h,R;

void circlePoint(int x,int y,int g,int h){
        glVertex2f(x+g,y+h);
        glVertex2f(-x+g,y+h);
        glVertex2f(x+g,-y+h);
        glVertex2f(-x+g,-y+h);
        glVertex2f(y+g,x+h);
        glVertex2f(y+g,-x+h);
        glVertex2f(-y+g,x+h);
        glVertex2f(-y+g,-x+h);

}
void myInit(void)
{
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 0.0, 1.0);

    glPointSize(3.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // setting window dimension in X- and Y- direction
    gluOrtho2D(-100, 100, -100, 100);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POINTS);
    glColor3f(0.0,1.0,0.0);
    for (int y = -500; y <= 500; y += 1){
            glVertex2f(0, y);
    }
    for (int x = -500; x <= 500; x += 1){
            glVertex2f(x, 0);
    }
    glEnd();
    int g= 0,h=0;
    int x=0,y=R,r=R;
    int di = 1 - r;
    glBegin(GL_POINTS);
    glColor3f(1,0,0);
    while(x<=y){
        // glVertex2f(x,y);
        circlePoint(x,y,g,h);
        if(di<0){
            di+= 2*x +3;
        }
        else{
            di+=2*(x-y)+5;
            y--;
        }
        x++;
    }

    glEnd();

    glFlush();
}

int main(int argc, char **argv)
{
    printf("Enter g(x coordinate of center): ");
    scanf("%d",&g);
    printf("Enter h(y coordinate of center): ");
    scanf("%d",&h);
    printf("Enter radius: ");
    scanf("%d",&R);

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(1000, 1000);
    glutInitWindowPosition(0, 0);

    // Giving name to window
    glutCreateWindow("Original ViewPort");
    myInit();

    // glutDisplayFunc(display);
    // glutInitWindowSize(600, 800);
    // glutInitWindowPosition(900, 0);
    // glutCreateWindow("Translated ViewPort");
    // myInit2();

    glutDisplayFunc(display);
    glutMainLoop();
}
```

Lab Assignment-1
Submitted By:
Name: Himesh Maniyar
ID: 2020UCP1776

Output:



```
^[[Ahimesh@Ubuntu22:~/Desktop/CG La./circle
Enter g(x coordinate of center): 13
Enter h(y coordinate of center): 24
Enter radius: 45
```



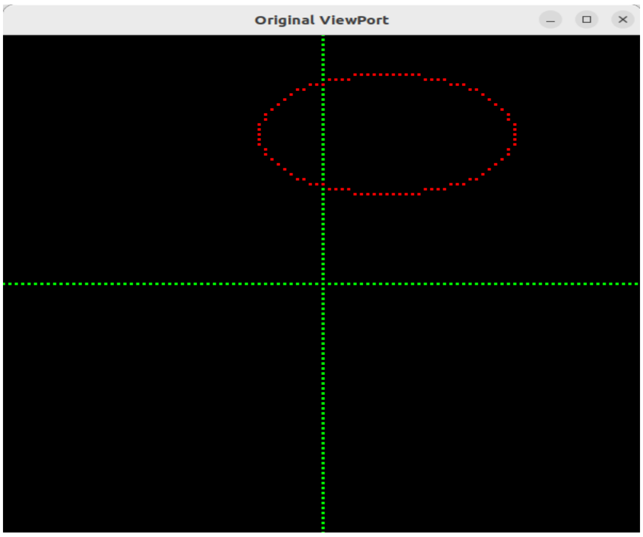
Lab Assignment-1
Submitted By:
Name: Himesh Maniyar
ID: 2020UCP1776

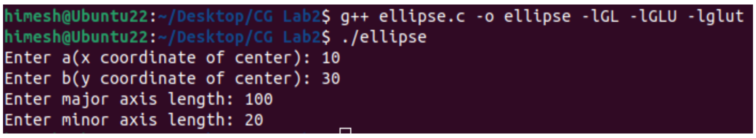
Program-3

Code:

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

int a,b,xc,yc;

void circlePoint(int x,int y){
        glVertex2f(x,y);
        glVertex2f(-x,y);
        glVertex2f(x,-y);
        glVertex2f(-x,-y);
        glVertex2f(y,x);
        glVertex2f(y,-x);
        glVertex2f(-y,x);
        glVertex2f(-y,-x);

}
void myInit(void)
{
    glClearColor(0.0, 0.0, 0.0, 1.0);
    glColor3f(0.0, 0.0, 1.0);

    glPointSize(3.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();

    // setting window dimension in X- and Y- direction
    gluOrtho2D(-50, 50, -50, 50);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_POINTS);

    glColor3f(0.0,1.0,0.0);
    for (int y = -400; y <= 400; y += 1)

        {
            glVertex2f(0, y);
        }
        for (int x = -400; x <= 400; x += 1)
        {
            glVertex2f(x, 0);
        }
    glEnd();
    int x = 0, y = b;
    int a2 = a * a;
    int b2 = b * b;
    int two_a2 = 2 * a2;
    int two_b2 = 2 * b2;
    int p = b2 - a2 * b + (a2 / 4);

    glBegin(GL_POINTS);
    glColor3f(1,0,0);
    glVertex2i(xc + x, yc + y);
    glVertex2i(xc - x, yc + y);
    glVertex2i(xc + x, yc - y);
    glVertex2i(xc - x, yc - y);

    while (a2 * y > x * b2)
    {
        if (p < 0)
        {
            x++;
            p += two_b2 * x + b2;
        }
        else
        {
            x++;
            y--;
            p += two_b2 * x - two_a2 * y + b2;
        }
        glVertex2i(xc + x, yc + y);
        glVertex2i(xc - x, yc + y);
        glVertex2i(xc + x, yc - y);
        glVertex2i(xc - x, yc - y);
    }

    p = b2 * (x + 0.5) * (x + 0.5) + a2 * (y - 1) * (y - 1) - a2 * b2;
    while (y >= 0)
    {
        if (p > 0)
        {
            y--;
            p -= two_a2 * y + a2;
        }
        else
        {
            x++;
            y--;
            p += two_b2 * x - two_a2 * y + a2;
        }
        glVertex2i(xc + x, yc + y);
        glVertex2i(xc - x, yc + y);
        glVertex2i(xc + x, yc - y);
        glVertex2i(xc - x, yc - y);
    }
    glEnd();

    glFlush();
}

int main(int argc, char **argv)
{
    printf("Enter x coordinate of center: ");
    scanf("%d",&xc);
    printf("Enter y coordinate of center: ");
    scanf("%d",&yc);
    printf("Enter major axis length: ");
    scanf("%d",&a);
    printf("Enter minor axis length: ");
    scanf("%d",&b);

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(600, 600);
    glutInitWindowPosition(0, 0);

    glutCreateWindow("Original ViewPort");
    myInit();

    glutDisplayFunc(display);
    glutMainLoop();
}
```

Output:





```
himesh@Ubuntu22:~/Desktop/CG Lab2$ g++ ellipse.c -o ellipse -lGL -lGLU -lglut
himesh@Ubuntu22:~/Desktop/CG Lab2$ ./ellipse
Enter a(x coordinate of center): 10
Enter b(y coordinate of center): 30
Enter major axis length: 100
Enter minor axis length: 20
```

Lab Assignment-1
Submitted By:
Name: Himesh Maniyar
ID: 2020UCP1776

Program-4

Code:

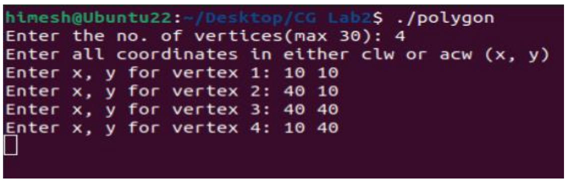
```
#include <GL/glut.h>
#include <stdio.h>

int x[30],y[30];
int n;

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

void initialiseVals(){
	glClear(GL_COLOR_BUFFER_BIT);
	glClearColor(0.0, 0.0, 0.0, 1.0);
	glColor3f(1.0, 0.0, 0.0);
	glMatrixMode(GL_PROJECTION);
	gluOrtho2D(-100, 100, -100, 100);
}


void bresenham(int a1, int b1, int a2, int b2){
	int x = a1;
	int y = b1;

	int dx = (a2 < a1) ? a1-a2 : a2-a1;
	int dy = (b2 < b1) ? b1-b2 : b2-b1;

	int incrementX =  (a2 < a1) ? -1 : 1;
	int incrementY =  (b2 < b1) ? -1 : 1;

	if (dx > dy){
		draw_pixel(x, y);
		int pi = 2*dy-dx;
		int changeD1 = 2*(dy-dx);
		int changeD2 = 2*dy;
		for (int i=0; i<dx; i++){
			if(pi<0)
				pi += changeD2;
			else{
				y += incrementY;
				pi += changeD1;
			}
			x += incrementX;
```

```
            draw_pixel(x, y);
        }

    }else{
        draw_pixel(x, y);
        int pi = 2*dx-dy;
        int changeD1 = 2*(dx-dy);
        int changeD2 = 2*dx;
        for (int i=0; i<dy; i++){
            if(pi<0)
                pi += changeD2;
            else{
                x += incrementX;
                pi += changeD1;
            }
            y += incrementY;
            draw_pixel(x, y);
        }
    }
    glFlush();
}


void polygon(){
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.0, 0.0, 1.0);
    glPointSize(4.0);
    glBegin(GL_LINES);
    glColor3f(0.0, 1.0, 0.0);
    glVertex2f(0, 1000);
    glVertex2f(0, -1000);
    glEnd();
    glBegin(GL_LINES);
    glColor3f(0.0, 1.0, 0.0);
    glVertex2f(1000, 0);
    glVertex2f(-1000, 0);
    glEnd();

    for(int i=0;i<(n-1);i++)
        bresenham(x[i],y[i],x[i+1],y[i+1]);

    bresenham(x[n-1], y[n-1], x[0], y[0]);

}

int main(int argc, char **argv){
    printf("Enter the no. of vertices(max 30): ");
    scanf("%d",&n);
```

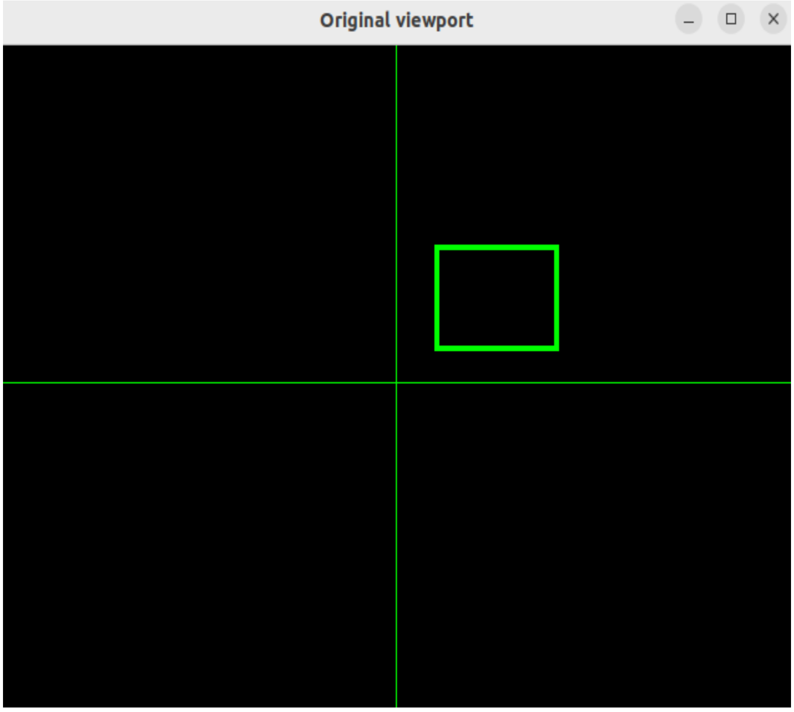
Lab Assignment-1
Submitted By:
Name: Himesh Maniyar
ID: 2020UCP1776

```
    printf("Enter all coordinates in either clw or acw (x, y)\n");
    for(int i=0; i<n; i++){
        printf("Enter x, y for vertex %d: ", (i+1));
        int temp1, temp2;
        scanf("%d",&temp1);
        scanf("%d",&temp2);
        x[i]=temp1;
        y[i]=temp2;
    }

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(10, 10);
    glutCreateWindow("Original viewport");
    initialiseVals();
    glutDisplayFunc(polygon);
    glutMainLoop();
    return 0;
}
```

Output:



```
himesh@Ubuntu22:~/Desktop/CG Lab2$ ./polygon
Enter the no. of vertices(max 30): 4
Enter all coordinates in either clw or acw (x, y)
Enter x, y for vertex 1: 10 10
Enter x, y for vertex 2: 40 10
Enter x, y for vertex 3: 40 40
Enter x, y for vertex 4: 10 40
```


Original viewport