**Name:** Nishan Paul

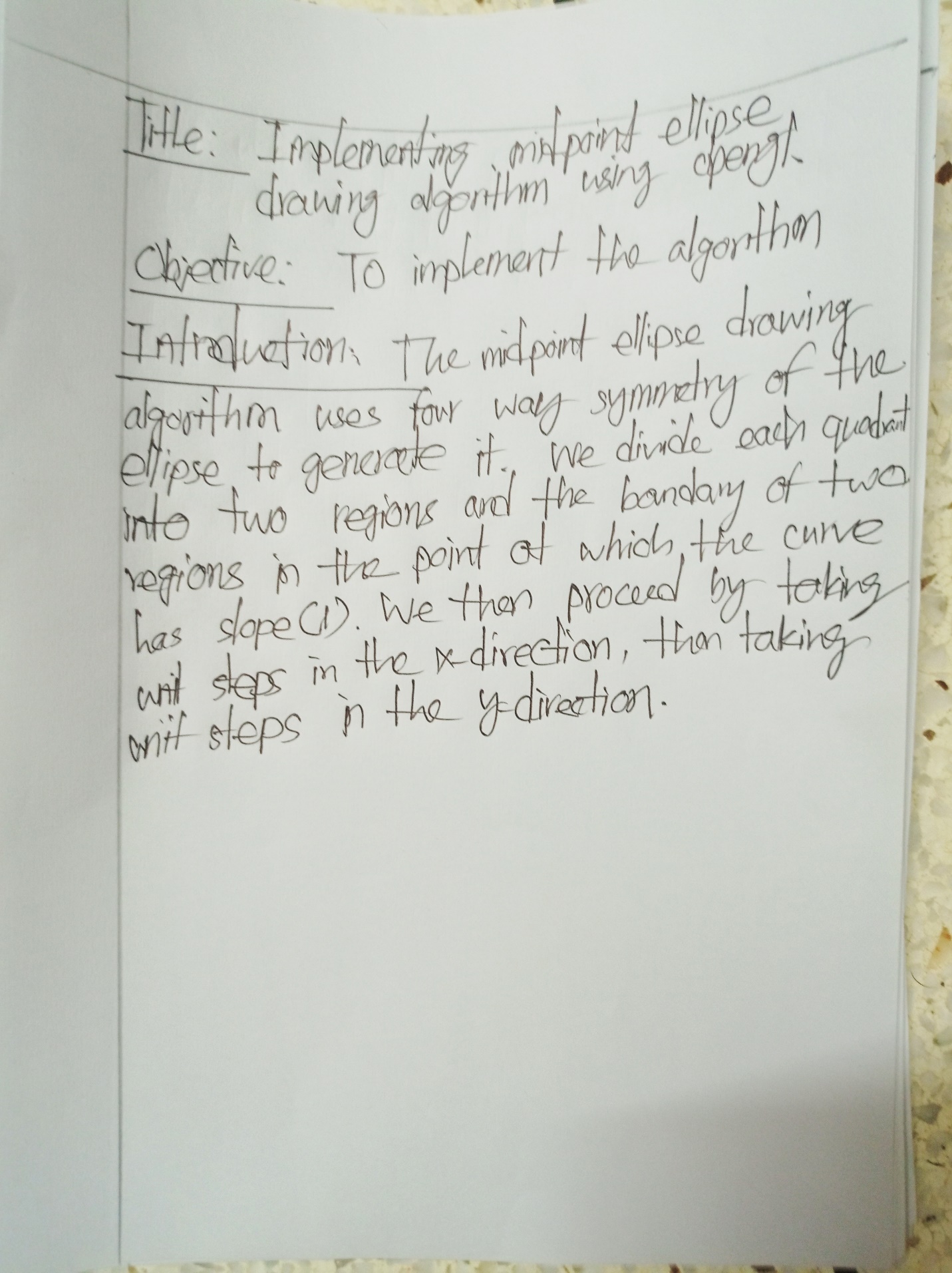
**ID:** 1604085

**Course Title:** Computer Graphics (Sessional)

**Course No:** CSE-458

**Level 4, Term 1**

**Assignment 05**



**Code:**

#include <windows.h>

#include <GL/glut.h>

#include<iostream>

using namespace std;

void algo\_rectangle(int x1, int y1, int x2, int y2) {

glBegin(GL\_POINTS);

glVertex2i(x1, y1);

int dx = abs(x2-x1);

int dy = abs(y2-y1);

int incx = x2<x1? -1 : 1;

int incy = y2<y1? -1 : 1;

int x = x1;

int y = y1;

int e = dx>dy? 2\*dy-dx : 2\*dx-dy;

int ifst = dx>dy? 2\*(dy-dx) : 2\*(dx-dy);

int isec = dx>dy? 2\*dy : 2\*dx;

int limit = dx>dy? dx:dy;

for(int i=0; i<limit; i++) {

if (e >= 0) {

if(dx > dy) y += incy;

if(dx <= dy) x += incx;

e += ifst;

}

else

e += isec;

if(dx > dy) x += incx;

if(dx <= dy) y += incy;

glVertex2i(x, y);

}

glEnd();

glFlush();

}

void algo\_ellipse\_(int center\_x, int center\_y, int x, int y) {

glVertex2i(center\_x + x, center\_y + y);

glVertex2i(center\_x - x, center\_y + y);

glVertex2i(center\_x + x, center\_y - y);

glVertex2i(center\_x - x, center\_y - y);

}

void algo\_ellipse(int center\_x, int center\_y, int a, int b) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glColor3f(1.0,0.0,0.0);

glBegin(GL\_POINTS);

float x = 0;

float y = b;

float p = b\*b - (a\*a\*b) + (a\*a\*0.25) ;

float dx = 0;

float dy = a\*a\*2\*b;

while(dx < dy) {

algo\_ellipse\_(center\_x, center\_y, x, y);

x++;

dx += (b\*b\*2);

if(p < 0)

p = p + dx + (b\*b);

else {

y--;

dy -= (a\*a\*2);

p += dx - dy +(b\*b);

}

}

float p2 = (b\*b\*(x + 0.5)\*(x + 0.5)) + (a\*a\*(y - 1)\*(y - 1)) - (a\*a\*b\*b);

while(y > 0) {

algo\_ellipse\_(center\_x, center\_y, x, y);

y--;

dy -= (a\*a\*2);

if(p2 >= 0)

p2 -= dy + (a\*a);

else {

x++;

dx += (2\*b\*b);

p2 += dx - dy + (a\*a);

}

}

glEnd();

glFlush();

}

void init(void) {

glClear(GL\_COLOR\_BUFFER\_BIT);

glClearColor(0,0,0,0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0, 640, 0, 480);

}

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

{

glutInit(&argc,argv);

glutInitWindowSize(1000, 1000);

glutInitWindowPosition(100, 100);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutCreateWindow("lab 5");

init();

glutDisplayFunc([]() {

algo\_ellipse(300, 300, 100, 70);

algo\_rectangle(220,260,380,260);

algo\_rectangle(380,260,380,340);

algo\_rectangle(380,340,220,340);

algo\_rectangle(220,340,220,260);

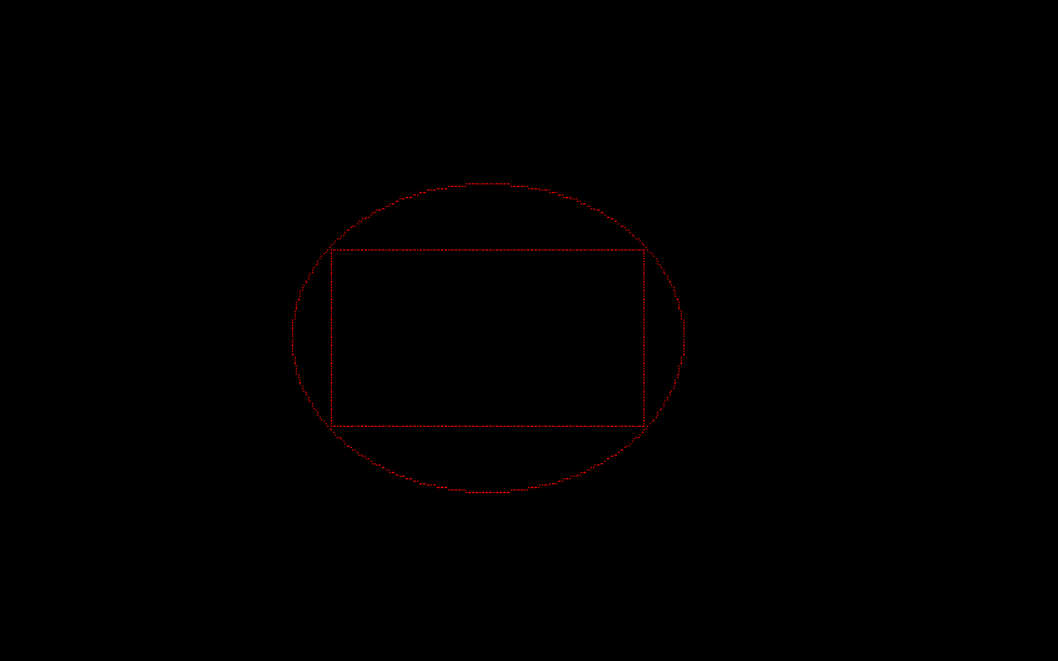
});

glutMainLoop();

return 0;

}

**Output:**

****

