**Name:** Nishan Paul

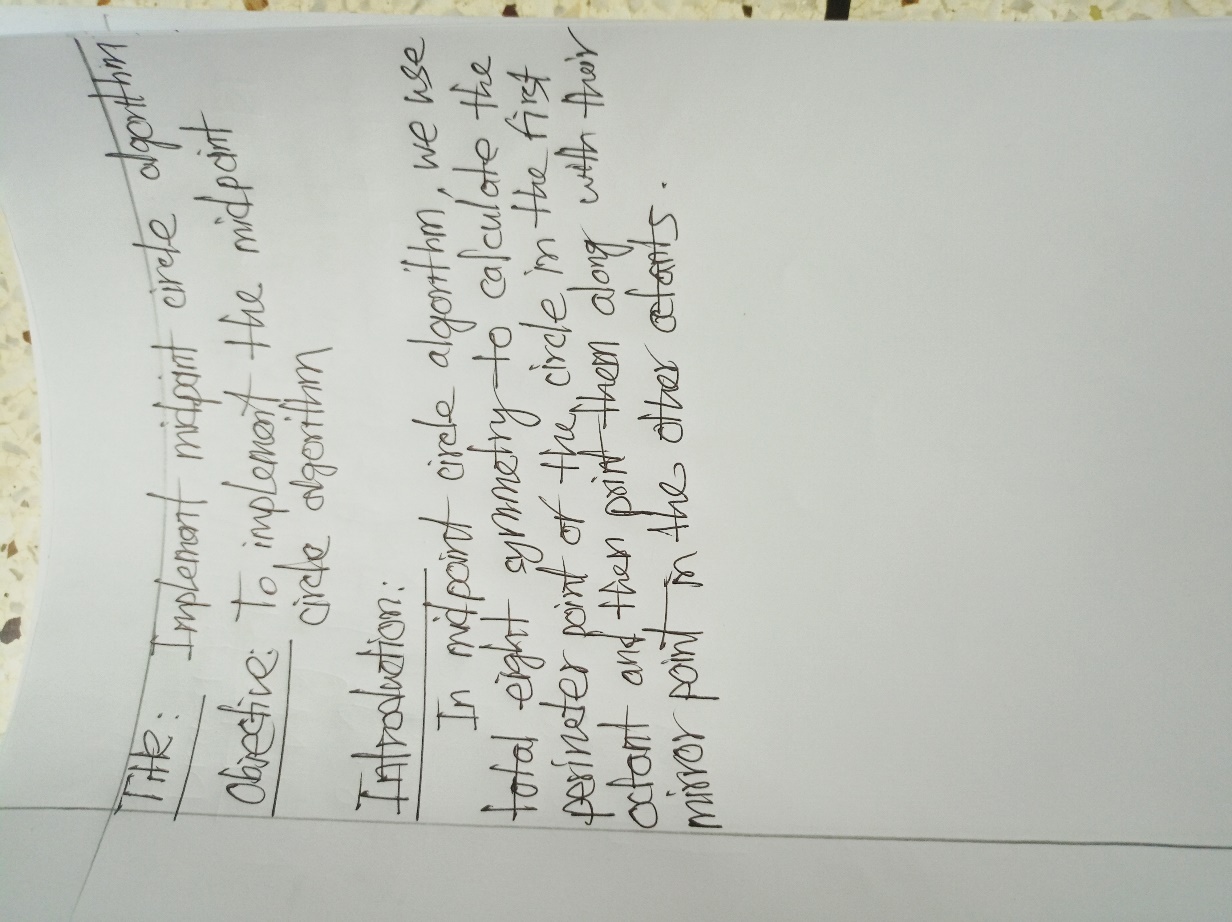
**ID:** 1604085

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

**Course No:** CSE-458

**Level 4, Term 1**

**Assignment 04**

****

**Code:**

#include<windows.h>

#include<GL/glut.h>

#include<stdlib.h>

#include<stdio.h>

using namespace std;

void algo\_1st(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\_2nd(int X1, int Y1, int r) {

glBegin(GL\_POINTS);

int x = 0, y = r;

float decision = 1 - r;

while(y > x) {

if(decision < 0) {

x++;

decision += 2\*x+3;

}

else {

y--;

x++;

decision += 2\*(x-y)+5;

}

glVertex2i(x+X1, y+Y1);

glVertex2i(x+X1, -y+Y1);

glVertex2i(-x+X1, y+Y1);

glVertex2i(-x+X1, -y+Y1);

glVertex2i(y+X1, x+Y1);

glVertex2i(-y+X1, x+Y1);

glVertex2i(y+X1, -x+Y1);

glVertex2i(-y+X1, -x+Y1);

}

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);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

glutInitWindowSize(1000,1000);

glutInitWindowPosition(100,100);

glutCreateWindow("3rd Task");

init();

glutDisplayFunc([]() {

algo\_1st(100,50,150,50);

algo\_1st(150,50,150,400);

algo\_1st(150,400,100,400);

algo\_1st(100,400,100,50);

algo\_2nd(125,425,25);

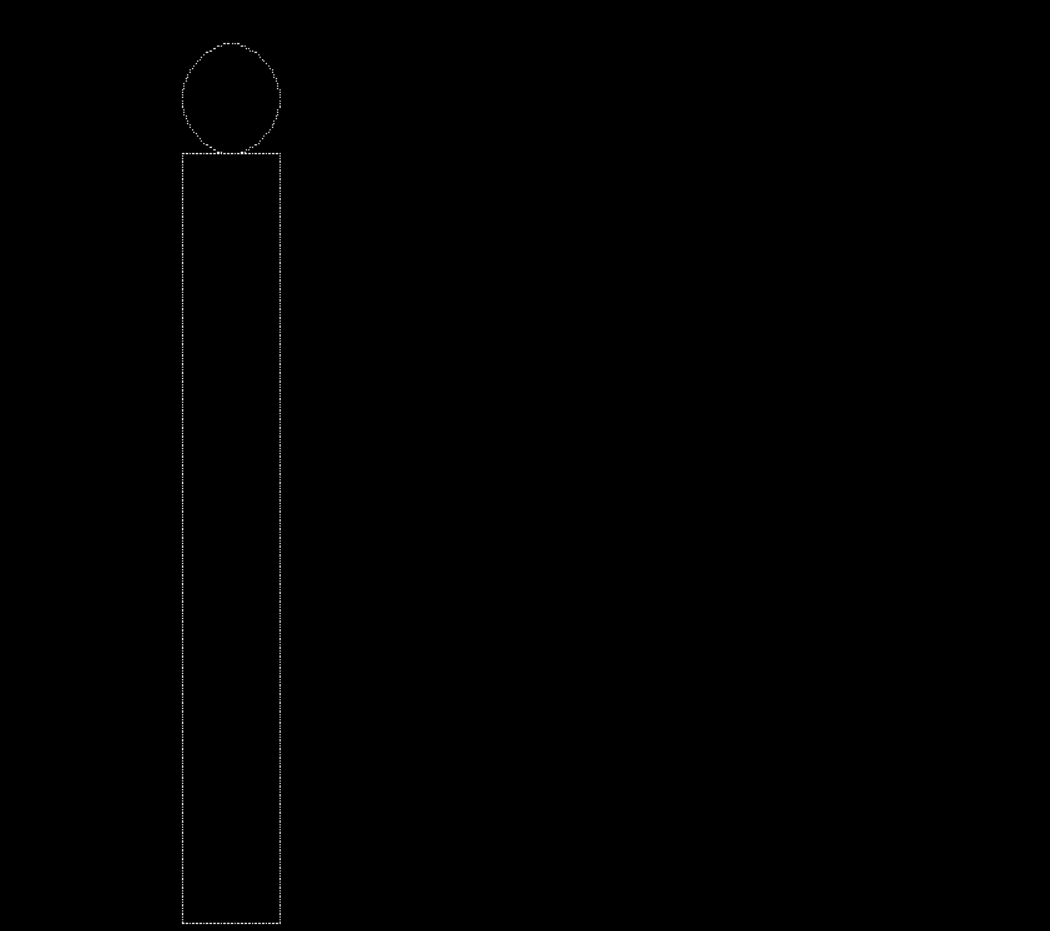
});

glutMainLoop();

return 0;

}

**Output:**

****

