#include<stdlib.h>

#include<stdio.h>

#include <GL/glut.h>

#include <GL/gl.h>

float x1, x2, y1, y2;

void DDA() {

int dy, dx, step, x, y, k, Xin, Yin;

dx = x2 - x1;

dy = y2 - y1;

if (abs(dx) > abs(dy)) {

step = abs(dx);

}

else

step = abs(dy);

Xin = dx / step;

Yin = dy / step;

x = x1;

y = y1;

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

for (k = 1; k <= step; k++) {

x = x + Xin;

y = y + Yin;

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

}

glFlush();

}

void Bresenham() {

GLint dx = x2 - x1;

GLint dy = y2 - y1;

GLint steps, k;

steps = dx;

GLint x, y, p0 = (2 \* dy) - dx;

glBegin(GL\_POINTS);

glVertex2i(x1, y1);

glEnd();

glFlush();

x = x1;

y = y1;

for (k = 0; k < steps; k++)

{

if (p0 < 0)

{

p0 = p0 + (2 \* dy);

x += 1;

}

else

{

p0 = p0 + (2 \* dy) - (2 \* dx);

x += 1;

y += 1;

}

glBegin(GL\_POINTS);

glVertex2i(x, y);

glEnd();

glFlush();

}

}

void display() {

//DDA();

Bresenham();

}

void init() {

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(640, 480);

glutInitWindowPosition(100, 150);

glutCreateWindow("");

}

void myInit(void) {

glClearColor(0.0, 0.0, 0.0, 0.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

gluOrtho2D(0.0, 640.0, 0.0, 480.0);

}

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

printf("Value of x1 : ");

scanf\_s("%f", &x1);

printf("Value of y1 : ");

scanf\_s("%f", &y1);

printf("Value of x2 : ");

scanf\_s("%f", &x2);

printf("Value of y2 : ");

scanf\_s("%f", &y2);

init();

myInit();

glutDisplayFunc(display);

glutMainLoop();

}