Name – Isha Kishor Katariya

Class - SEIT Div -A

Roll No – 207A056

Experiment - Implement DDA and Bresenham line drawing algorithm to draw:

i) Simple Line ii) Dotted Line iii)Dashed Line iv) Solid line; using mouse interface. Divide the screen in four quadrants with center as (0, 0). The line should work for all the slopes positive as well as negative.

#include<GL/glut.h> #include <string> void init() { glClearColor(0,

0, 0, 0);

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0, 500, 0, 500);

} void displayPoint(float x, float y) { glColor3f(1,

1, 0);

glBegin(GL\_POINTS); glVertex2f(x, y); glEnd(); } void drawLine(int x1, int y1, int x2, int y2,char line = ’A’) { int P; int dx = x2 - x1; int dy = y2 - y1; int x = x1; int y = y1; int x\_change = x1 > x2 ? -1 : 1; int y\_change = y1 > y2

? -1 :

1; displayPoint(x, y); if (dx == 0) { glBegin(GL\_POINTS);

for(int i=0; i < abs(dy); i++) { y += y\_change; if

(line == ’B’) { if (i % 3 == 0) { glVertex2f(x, y);

} } else if(line == ’C’) { if (i % 7 == 0) { glVertex2f(x, y);

}} else {

glVertex2f(x, y);

} }

glEnd();glFlush(); return; } if (dy == 0) {

glBegin(GL\_POINTS);

for(int i=0; i < abs(dx); i++) { x += x\_change; if (line == ’B’) { if (i % 3 == 0) { glVertex2f(x, y); } } else if(line == ’C’) { if (i % 7 == 0) { glVertex2f(x, y);

}

} else { glVertex2f(x, y); } } glEnd(); glFlush(); return; }

glBegin(GL\_POINTS);

if (dx > dy) { P = 2 \* dy - dx; for (int i = 0; i < dx; i++) { if (P > 0) { y += y\_change; P += 2 \* dy - 2 \* dx;

} else { P += 2 \* dy;

} x += x\_change; if

(line == ’B’) { if (i % 3 == 0) { glVertex2f(x, y); }

} else if(line == ’C’) { if (i % 7 == 0) { glVertex2f(x, y); } }else { glVertex2f(x, y);}

} } else { P = 2 \* dx - dy; for

(int i = 0; i < dy; i++) { if

(P > 0) { x += x\_change; P

+=

2 \* dx - 2 \* dy; } else { P += 2 \* dx; } y += y\_change; if (line == ’B’) { if (i % 3 == 0) { glVertex2f(x, y);

}

}

else if(line == ’C’) { if (i % 7 == 0) { glVertex2f(x, y);

} } else { glVertex2f(x, y); }

} } glEnd(); glFlush(); } void display() {

glClear(GL\_COLOR\_BUFFER\_BIT);

// A = SIMPLE LINE

// B = DOTTED LINE // C = DASHED LINE

drawLine(10, 10, 490, 490, ’A’); drawLine(10,

250, 490, 250, ’B’); drawLine(250, 490, 250,

10, ’C’); } int main(int argc, char\*\* argv) { glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB); glutInitWindowPosition(400, 100); glutInitWindowSize(500, 500); glutCreateWindow("Line Drawing");init(); glutDisplayFunc(display);glutMainLoop();

return 0;

}