EX 3

205001085
SABARIVASAN V

AIM:

To plot points that make up the line with endpoints (x0,y0) and (xn,yn) using Bresenham's line drawing algorithm.

Case 1: +ve slope Left to Right line

Case 2: +ve slope Right to Left line

Case 3: -ve slope Left to Right line

Case 4: -ve slope Right to Left line Each case has two subdivisions

(i) |m|<= 1 (ii) |m|>1
Note that all four cases of line drawing must be given as test cases.

ALGORITHM:

Input Line Endpoints,(x0,y0)and(xn,yn)
Load (x0,y0) into the frame buffer that is first point Calculate The Constants X,y,2y and 2y-2x calculate parameter p0 = 2 y - x Set pixel At Position(x0,y0) repeat the following steps until(xn,yn)is reached:

if pk < 0
set the next pixel at position (xk +1, yk) calculate new pk+1 = pk + 2 y

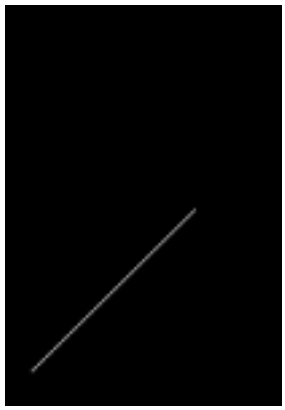
if pk ≥ 0
set the next pixel at position (xk +1, yk + 1) calculate new pk+1 = pk + 2(y - x)
Repeat last step x times.

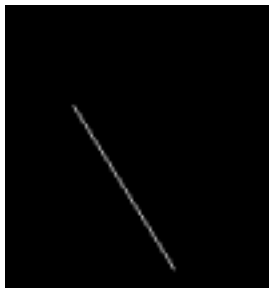
CODE :

```
#include <GLUT/glut.h>
#include <stdio.h>
#include<math.h>
int xstart, ystart, xend, yend;
void myInit() {
glClear(GL_COLOR_BUFFER_BIT);
glClearColor(0.0, 0.0, 0.0, 1.0);
glMatrixMode(GL_PROJECTION);
gluOrtho2D(0, 500, 0, 500);
}
void draw_pixel(int x, int y) {
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
}
void draw_line(int xstart, int xend, int ystart, int yend) { int dx, dy, i, e;
int incx, incy, inc1, inc2;
int x,y;
dx = abs(xend-xstart);
dy = abs(yend-ystart);
incx = 1;
if (xend < xstart) incx = -1;
incy = 1;
if (yend < ystart) incy = -1;
x = xstart;
y = ystart;
if (dx > dy) {
draw_pixel(x, y);
e = 2 * dy-dx;
inc1 = 2*(dy-dx);
inc2 = 2*dy;
for (i=0; i<dx; i++) {
if (e >= 0) {
y += incy;
e += inc1;
}
```

```
else
e += inc2;
x += incx;
draw_pixel(x, y);
}
} else {
draw_pixel(x, y);
e = 2*dx-dy;
inc1 = 2*(dx-dy);
inc2 = 2*dx;
for (i=0; i<dy; i++) {
if (e >= 0) {
x += incx;
e += inc1;
}
else
e += inc2;
y += incy;
draw_pixel(x, y);
}
}
}
void myDisplay() {
draw_line(xstart, xend, ystart, yend);
glFlush();
}
int main(int argc, char **argv) {
printf( "Enter (xstart, ystart, xend, yend)\n"); scanf("%d %d %d %d", &xstart,
&ystart, &xend, &yend); glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); glutInitWindowSize(500,
500);
glutInitWindowPosition(0, 0);
glutCreateWindow("Excercise 3");
myInit();
glutDisplayFunc(myDisplay);
glutMainLoop();
}
```

SAMPLE I/O:





LEARNING OUTCOME:

I learnt how to use bresenham's line drawing algorithm in c++ using the openGL library to draw a line.