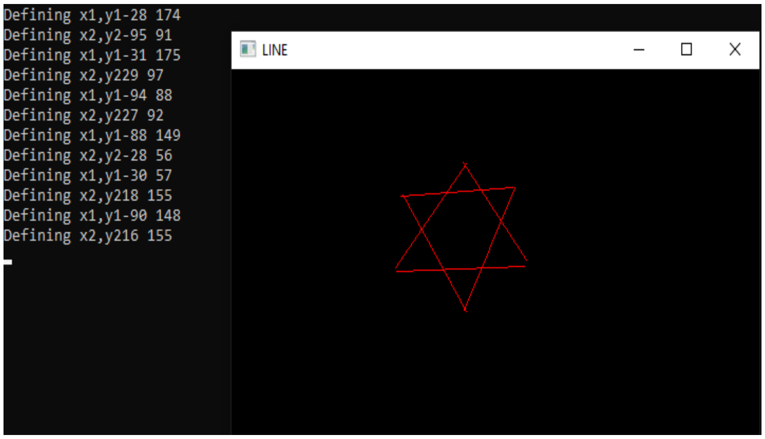
1. Write a program to generate a line using Bresenham's line drawing technique. Consider slopes greater than one and slopes less than one. User must able to draw as many lines and specify inputs through keyboard/mouse.

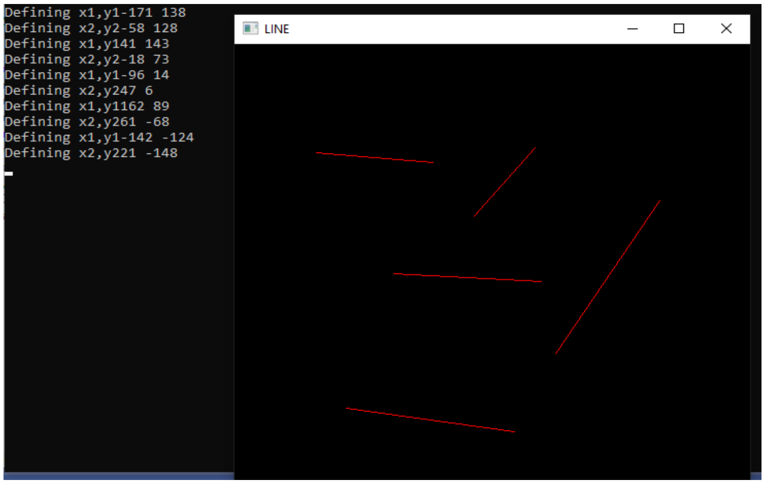
```
#include <iostream>
#include <GL/glut.h>
#include <time.h>
using namespace std;
int x1, x2, yc1, y2;
int flag = 0;
void draw_pixel(int x, int y)
{
        glColor3f(1, 0, 0);
        glBegin(GL_POINTS);
        glVertex2i(x, y);
        glEnd();
        glFlush();
}
void draw_line()
{
        int dx, dy, i, e;
        int incx, incy, inc1, inc2;
        int x, y;
        dx = x2 - x1;
        dy = y2 - yc1;
        if (dx < 0)dx = -dx;
        if (dy < 0)dy = -dy;
        incx = 1;
        if (x2 < x1)
                incx = -1;
        incy = 1;
        if (y2 < yc1)
                incy = -1;
        x = x1;
        y = yc1;
        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);
                }
        }
        glFlush();
}
void myinit()
{
        glClear(GL_COLOR_BUFFER_BIT);
        glClearColor(1, 1, 1, 1);
        gluOrtho2D(-250, 250, -250, 250);
}
void MyMouse(int button, int state, int x, int y)
{
        switch (button)
        {
        case GLUT_LEFT_BUTTON:
                if (state == GLUT_DOWN)
                {
                        if (flag == 0)
                        {
                                printf("Defining x1,y1");
                                x1 = x - 250;
                                yc1 = 250 - y;
                                flag++;
                                cout << x1 << " " << yc1 << " \n";

                        }
                        else
                        {
                                printf("Defining x2,y2");
                                x2 = x - 250;
                                y2 = 250 - y;
                                flag = 0;
                                cout << x2 << " " << y2 << " \n";
                                draw_line();
                        }
                }
                break;
        }
}
void display()
```

```
{}
int main(int ac, char* av[])
{
        /*
        //FOR KEYBOARD
        cout<<"X1\n";
        cin>>x1;
        cout<<"Y1\n";
        cin>>yc1;
        cout<<"X2\n";
        cin>>x2;
        cout<<"Y2\n";
        cin>>y2;
        //END KEYBOARD
        */
        glutInit(&ac, av);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(500, 500);
        glutInitWindowPosition(100, 200);
        glutCreateWindow("LINE");
        myinit();
        glutMouseFunc(MyMouse); //INCLUDE TO USE MOUSE, REMOVE WHILE USING KEYBOARD
                //draw_line(); //INCLUDE TO USE KEYBOARD, REMOVE WHILE USING MOUSE
                glutDisplayFunc(display);
        glutMainLoop();
}
```

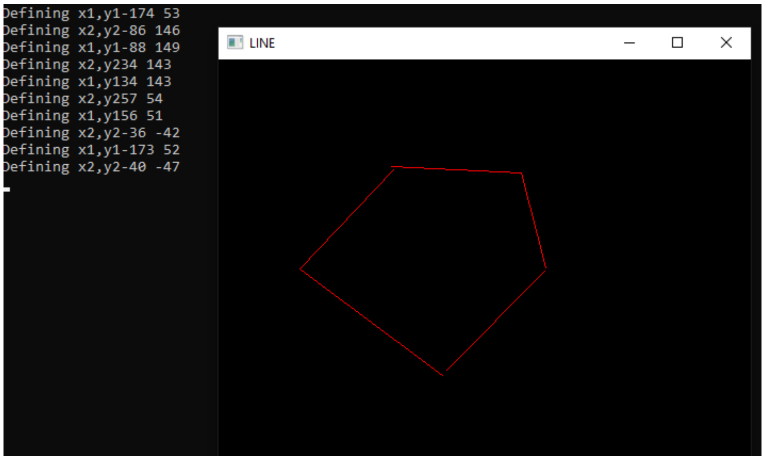
OUTPUT



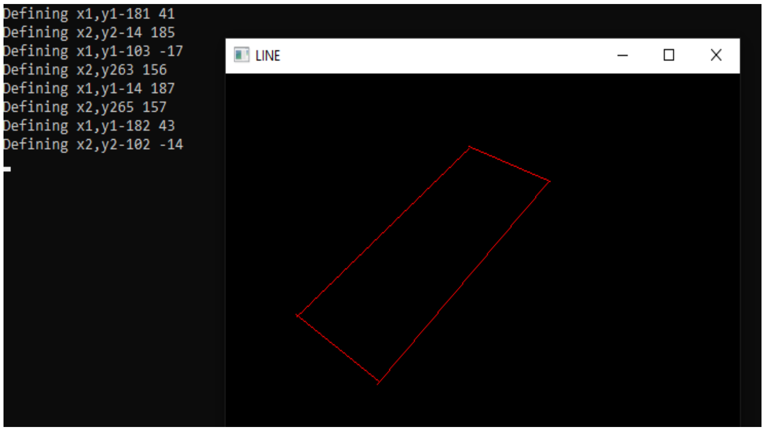
```
Defining x1,y1-28 174
Defining x2,y2-95 91
Defining x1,y1-31 175
Defining x2,y229 97
Defining x1,y1-94 88
Defining x2,y227 92
Defining x1,y1-88 149
Defining x2,y2-28 56
Defining x1,y1-30 57
Defining x2,y218 155
Defining x1,y1-90 148
Defining x2,y216 155
```



```
Defining x1,y1-171 138
Defining x2,y2-58 128
Defining x1,y141 143
Defining x2,y2-18 73
Defining x1,y1-96 14
Defining x2,y247 6
Defining x1,y1162 89
Defining x2,y261 -68
Defining x1,y1-142 -124
Defining x2,y221 -148
```



```
Defining x1,y1-174 53
Defining x2,y2-86 146
Defining x1,y1-88 149
Defining x2,y234 143
Defining x1,y134 143
Defining x2,y257 54
Defining x1,y156 51
Defining x2,y2-36 -42
Defining x1,y1-173 52
Defining x2,y2-40 -47
```



```
Defining x1,y1-181 41
Defining x2,y2-14 185
Defining x1,y1-103 -17
Defining x2,y263 156
Defining x1,y1-14 187
Defining x2,y265 157
Defining x1,y1-182 43
Defining x2,y2-102 -14
```