Name	:	**V Vikram**	**Class**	:	CSE ‘C’
Reg. No.	:	**18 5001 194**	**Date**	:	04/08/2021
Subject	:	**UCS1712---Graphics and Multimedia Lab**			

QUESTION :

Lab Exercise 3 :

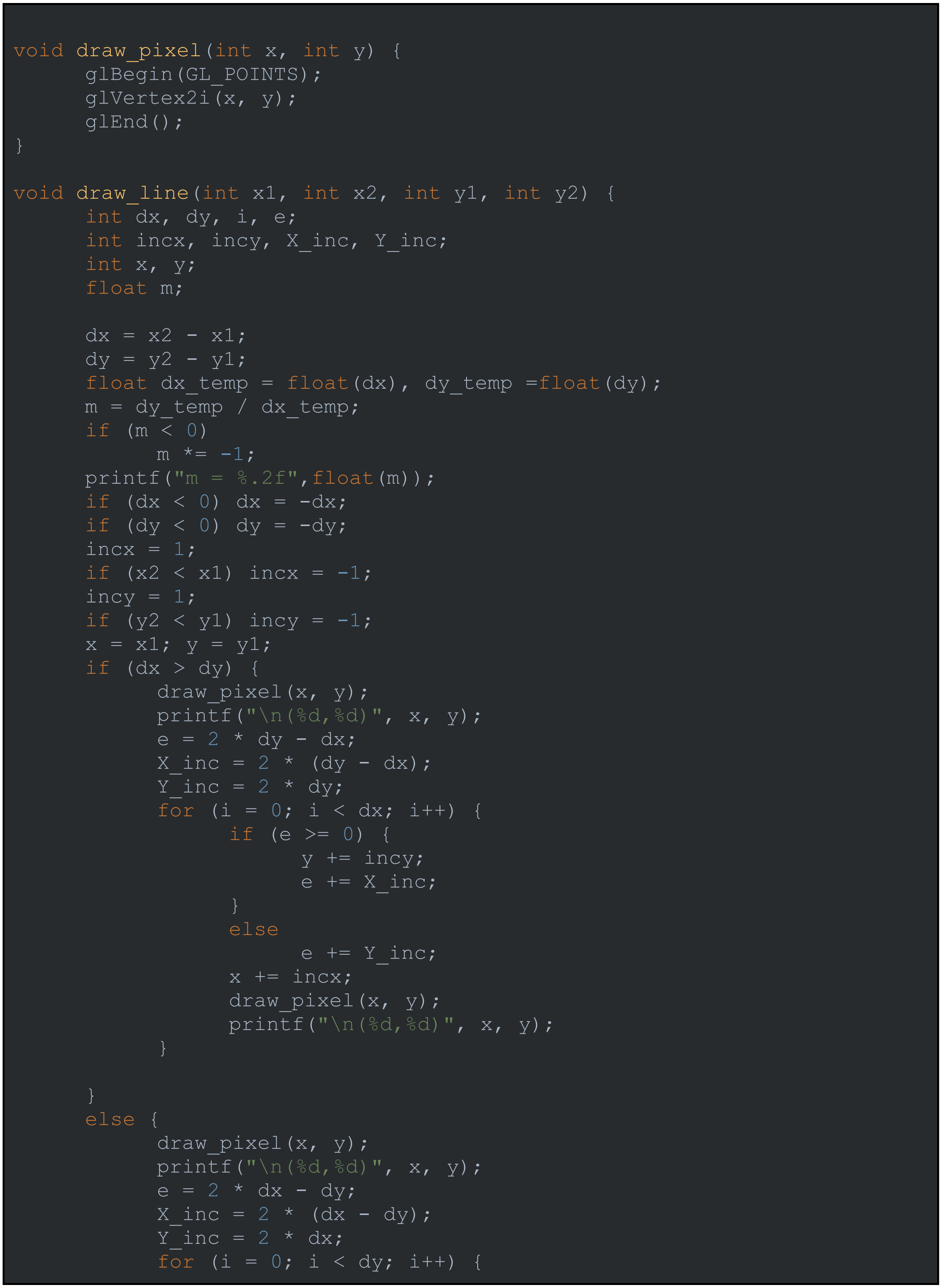
Bresenham’s Line Drawing Algorithm in C++ using OpenGL

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

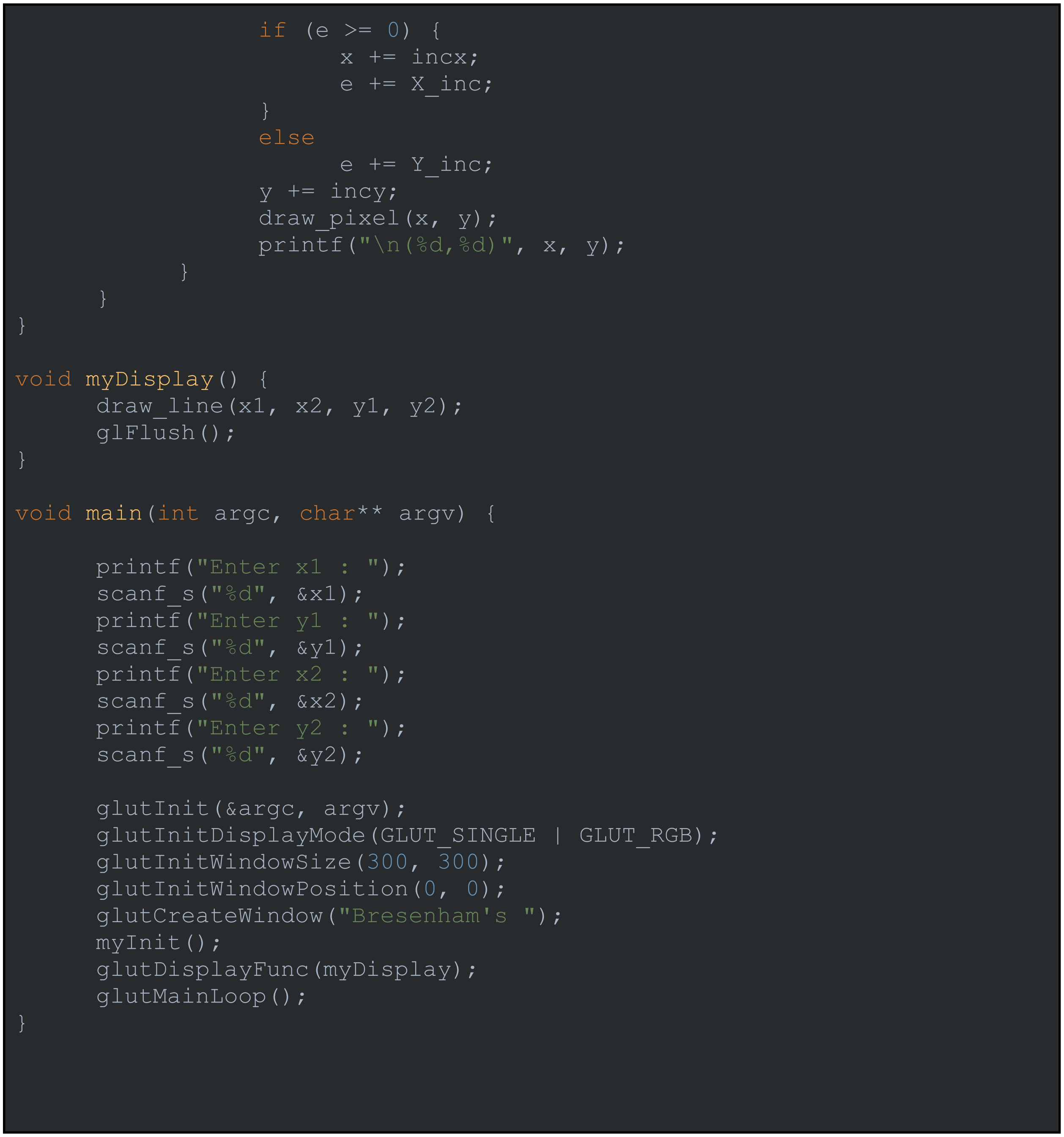
Case 1 : +ve slope Left to Right line
Case 2 : +ve slope Right to Left line
Case 3 : -ve slope Left to Right line
Case 4 : -ve slope Right to Left line
Each case has two subdivisions
(i) |m|<= 1 (ii) |m|>1

CODE :

```
#include <gl/glut.h>
#include <stdio.h>

int x1, y1, x2, y2;

void myInit() {
	glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
	glClearColor(0.0, 0.0, 0.0, 0.0);
	glColor3f(255.0f / 255.0f, 255.0f / 255.0f, 255.0f / 255.0f);
	glPointSize(2.0f);
	glEnable(GL_DEPTH_TEST);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(-100, 100, -100, 100);
}
```



```
void draw_pixel(int x, int y) {
      glBegin(GL_POINTS);
      glVertex2i(x, y);
      glEnd();
}

void draw_line(int x1, int x2, int y1, int y2) {
      int dx, dy, i, e;
      int incx, incy, X_inc, Y_inc;
      int x, y;
      float m;

      dx = x2 - x1;
      dy = y2 - y1;
      float dx_temp = float(dx), dy_temp =float(dy);
      m = dy_temp / dx_temp;
      if (m < 0)
            m *= -1;
      printf("m = %.2f",float(m));
      if (dx < 0) dx = -dx;
      if (dy < 0) dy = -dy;
      incx = 1;
      if (x2 < x1) incx = -1;
      incy = 1;
      if (y2 < y1) incy = -1;
      x = x1; y = y1;
      if (dx > dy) {
            draw_pixel(x, y);
            printf("\n(%d,%d)", x, y);
            e = 2 * dy - dx;
            X_inc = 2 * (dy - dx);
            Y_inc = 2 * dy;
            for (i = 0; i < dx; i++) {
                  if (e >= 0) {
                        y += incy;
                        e += X_inc;
                  }
                  else
                        e += Y_inc;
                  x += incx;
                  draw_pixel(x, y);
                  printf("\n(%d,%d)", x, y);
            }

      }
      else {
            draw_pixel(x, y);
            printf("\n(%d,%d)", x, y);
            e = 2 * dx - dy;
            X_inc = 2 * (dx - dy);
            Y_inc = 2 * dx;
            for (i = 0; i < dy; i++) {
```

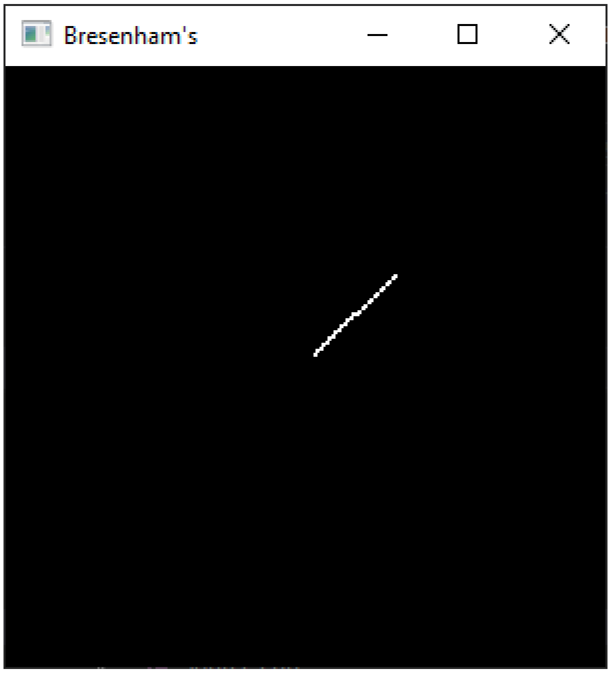


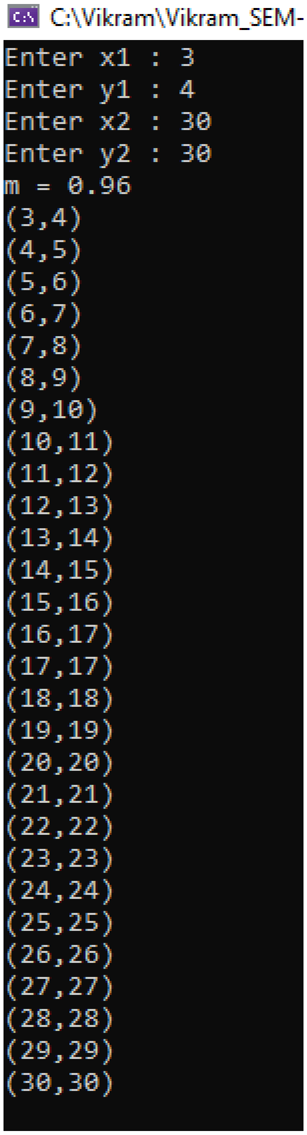
```
            if (e >= 0) {
                x += incx;
                e += X_inc;
            }
            else
                e += Y_inc;
            y += incy;
            draw_pixel(x, y);
            printf("\n(%d,%d)", x, y);
        }
    }
}

void myDisplay() {
    draw_line(x1, x2, y1, y2);
    glFlush();
}

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

    printf("Enter x1 : ");
    scanf_s("%d", &x1);
    printf("Enter y1 : ");
    scanf_s("%d", &y1);
    printf("Enter x2 : ");
    scanf_s("%d", &x2);
    printf("Enter y2 : ");
    scanf_s("%d", &y2);

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(300, 300);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Bresenham's ");
    myInit();
    glutDisplayFunc(myDisplay);
    glutMainLoop();
}
```

OUTPUT SNAPSHOTS :-

+ve slope left to right:

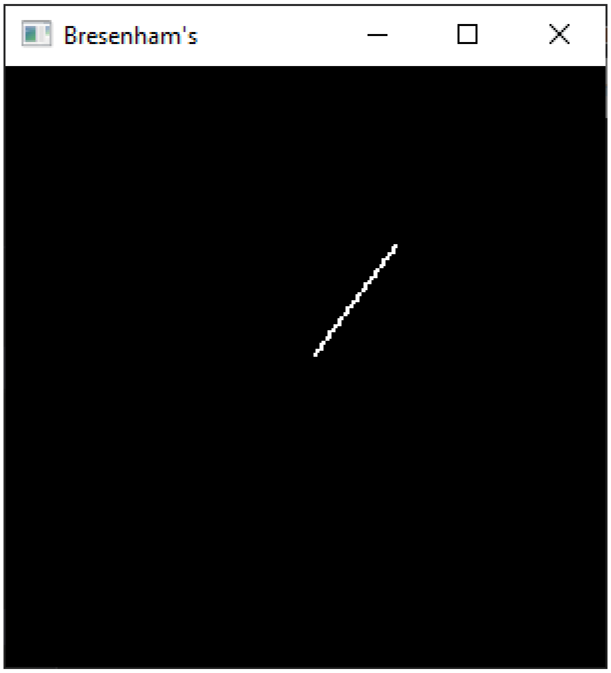
m<=1

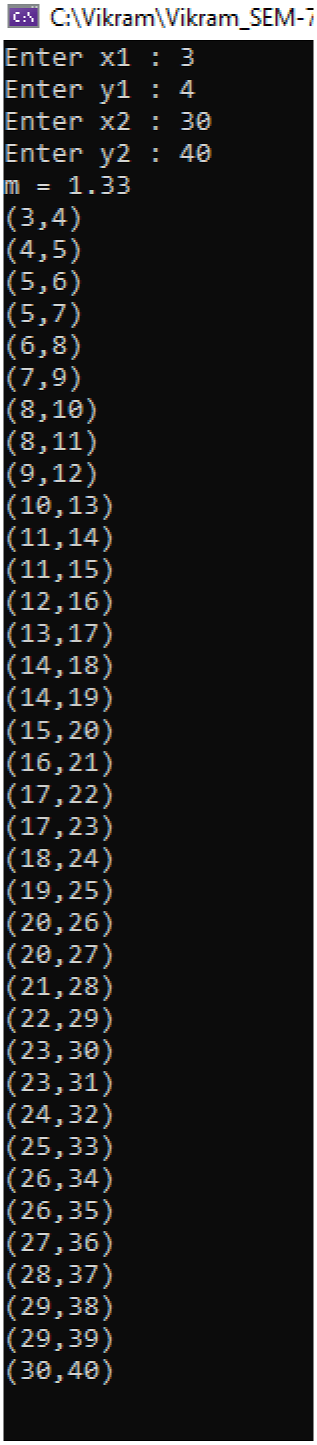




```
Enter x1 : 3
Enter y1 : 4
Enter x2 : 30
Enter y2 : 30
m = 0.96
(3,4)
(4,5)
(5,6)
(6,7)
(7,8)
(8,9)
(9,10)
(10,11)
(11,12)
(12,13)
(13,14)
(14,15)
(15,16)
(16,17)
(17,17)
(18,18)
(19,19)
(20,20)
(21,21)
(22,22)
(23,23)
(24,24)
(25,25)
(26,26)
(27,27)
(28,28)
(29,29)
(30,30)
```

m>1 :

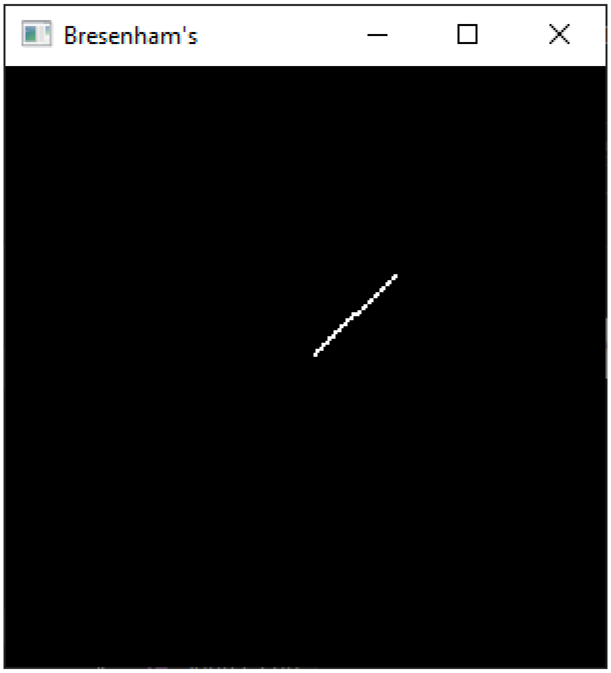


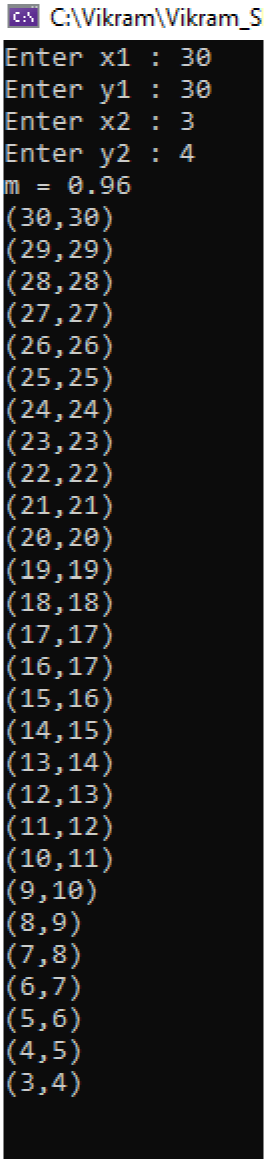


```
Enter x1 : 3
Enter y1 : 4
Enter x2 : 30
Enter y2 : 40
m = 1.33
(3,4)
(4,5)
(5,6)
(5,7)
(6,8)
(7,9)
(8,10)
(8,11)
(9,12)
(10,13)
(11,14)
(11,15)
(12,16)
(13,17)
(14,18)
(14,19)
(15,20)
(16,21)
(17,22)
(17,23)
(18,24)
(19,25)
(20,26)
(20,27)
(21,28)
(22,29)
(23,30)
(23,31)
(24,32)
(25,33)
(26,34)
(26,35)
(27,36)
(28,37)
(29,38)
(29,39)
(30,40)
```

+ve slope right to left :

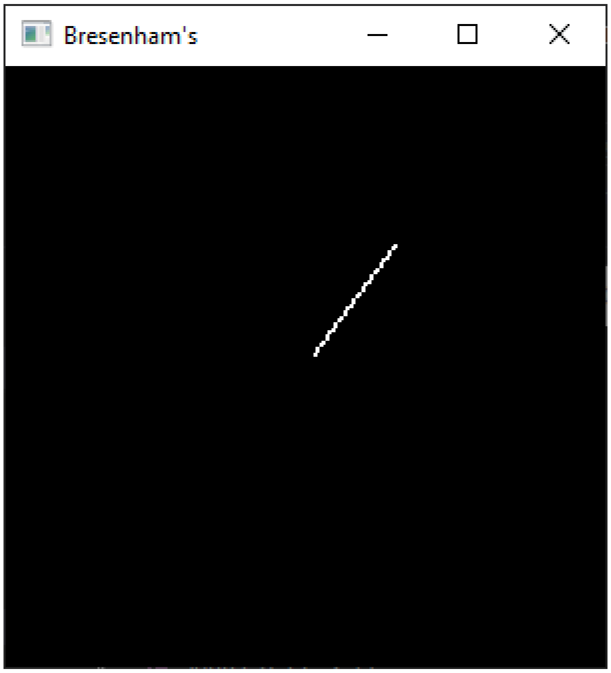
m<=1:

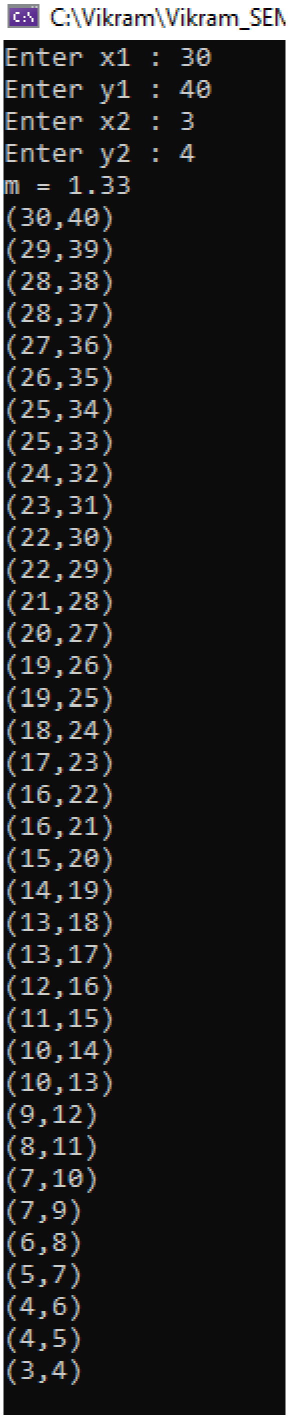




```
Enter x1 : 30
Enter y1 : 30
Enter x2 : 3
Enter y2 : 4
m = 0.96
(30,30)
(29,29)
(28,28)
(27,27)
(26,26)
(25,25)
(24,24)
(23,23)
(22,22)
(21,21)
(20,20)
(19,19)
(18,18)
(17,17)
(16,17)
(15,16)
(14,15)
(13,14)
(12,13)
(11,12)
(10,11)
(9,10)
(8,9)
(7,8)
(6,7)
(5,6)
(4,5)
(3,4)
```

m>1:

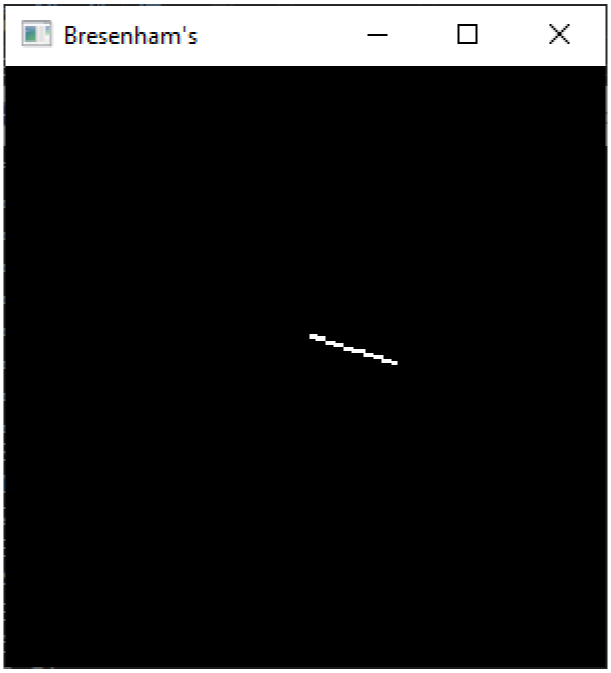


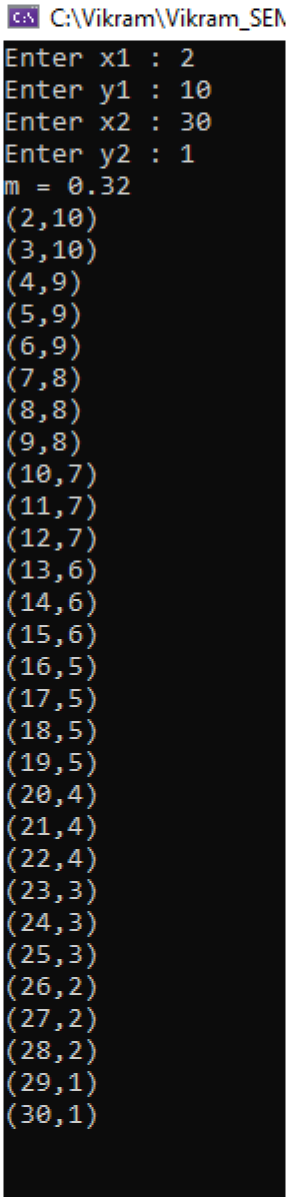


```
Enter x1 : 30
Enter y1 : 40
Enter x2 : 3
Enter y2 : 4
m = 1.33
(30,40)
(29,39)
(28,38)
(28,37)
(27,36)
(26,35)
(25,34)
(25,33)
(24,32)
(23,31)
(22,30)
(22,29)
(21,28)
(20,27)
(19,26)
(19,25)
(18,24)
(17,23)
(16,22)
(16,21)
(15,20)
(14,19)
(13,18)
(13,17)
(12,16)
(11,15)
(10,14)
(10,13)
(9,12)
(8,11)
(7,10)
(7,9)
(6,8)
(5,7)
(4,6)
(4,5)
(3,4)
```

-ve slope left to right:

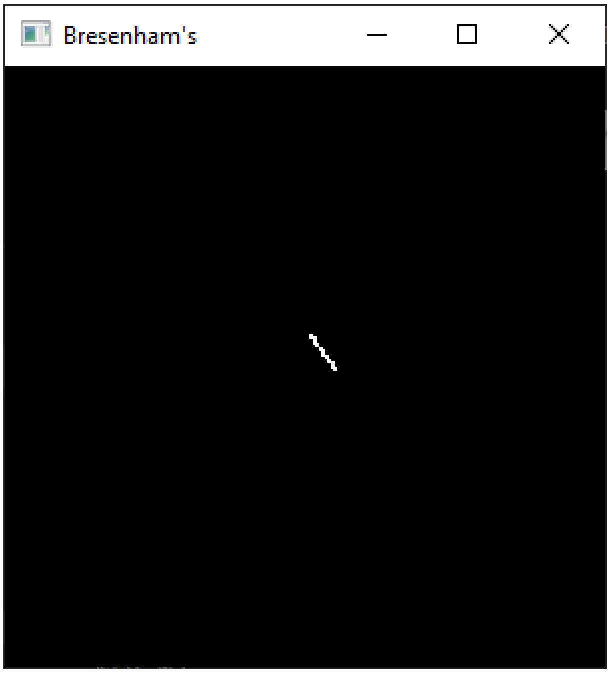
|m|<=1 :

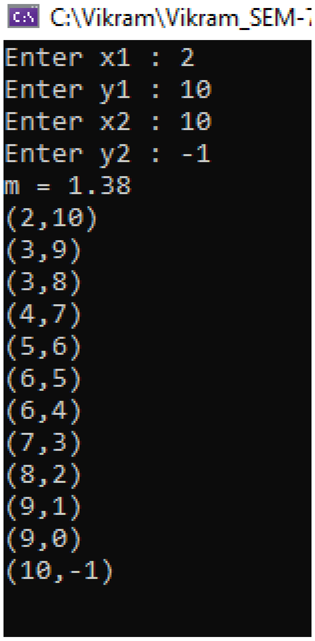




```
Enter x1 : 2
Enter y1 : 10
Enter x2 : 30
Enter y2 : 1
m = 0.32
(2,10)
(3,10)
(4,9)
(5,9)
(6,9)
(7,8)
(8,8)
(9,8)
(10,7)
(11,7)
(12,7)
(13,6)
(14,6)
(15,6)
(16,5)
(17,5)
(18,5)
(19,5)
(20,4)
(21,4)
(22,4)
(23,3)
(24,3)
(25,3)
(26,2)
(27,2)
(28,2)
(29,1)
(30,1)
```

|m|>1:

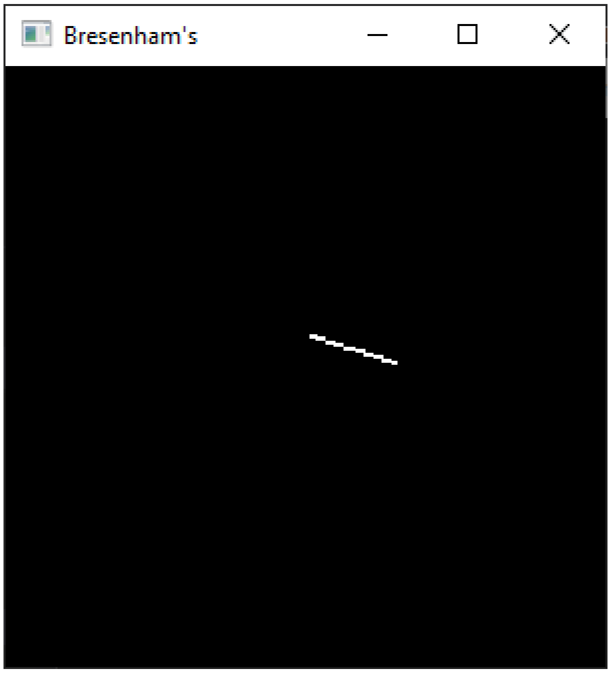


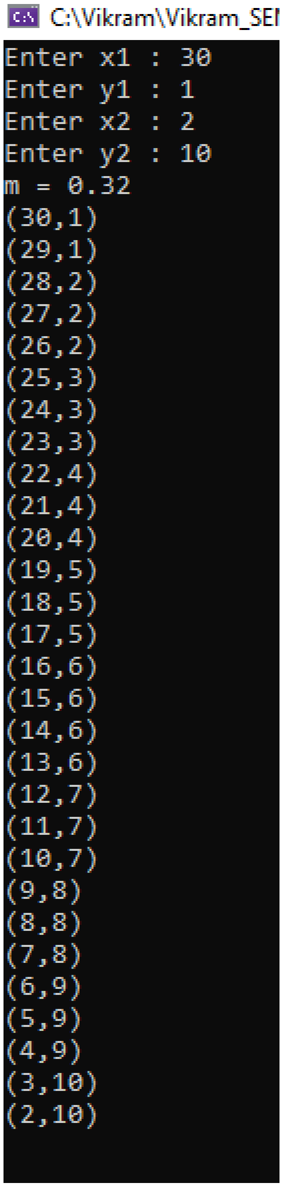


```
Enter x1 : 2
Enter y1 : 10
Enter x2 : 10
Enter y2 : -1
m = 1.38
(2,10)
(3,9)
(3,8)
(4,7)
(5,6)
(6,5)
(6,4)
(7,3)
(8,2)
(9,1)
(9,0)
(10,-1)
```

-ve slope right to left :

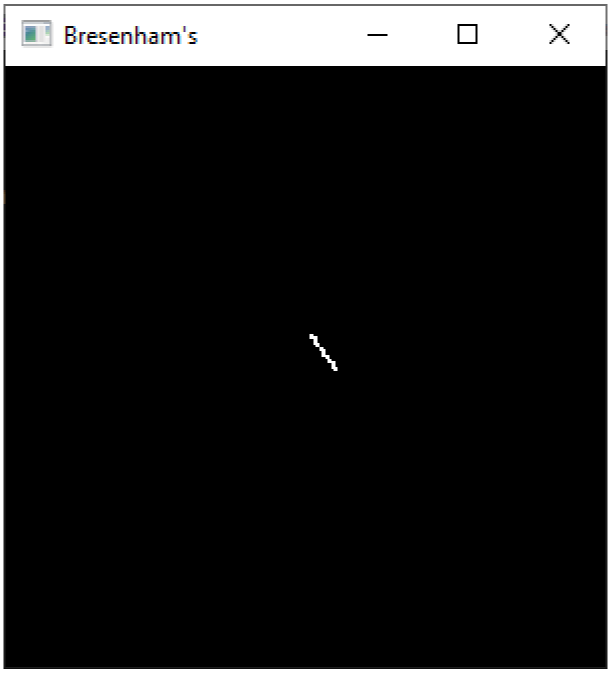
|m|<=1 :

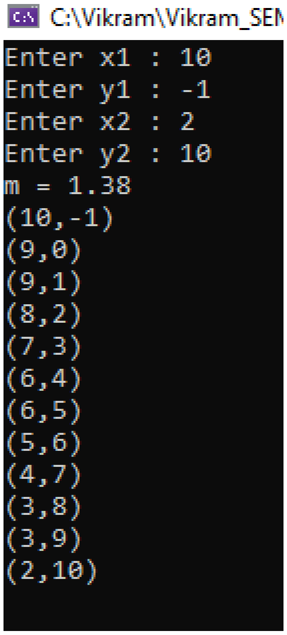




```
Enter x1 : 30
Enter y1 : 1
Enter x2 : 2
Enter y2 : 10
m = 0.32
(30,1)
(29,1)
(28,2)
(27,2)
(26,2)
(25,3)
(24,3)
(23,3)
(22,4)
(21,4)
(20,4)
(19,5)
(18,5)
(17,5)
(16,6)
(15,6)
(14,6)
(13,6)
(12,7)
(11,7)
(10,7)
(9,8)
(8,8)
(7,8)
(6,9)
(5,9)
(4,9)
(3,10)
(2,10)
```

|m|>1 :





```
Enter x1 : 10
Enter y1 : -1
Enter x2 : 2
Enter y2 : 10
m = 1.38
(10,-1)
(9,0)
(9,1)
(8,2)
(7,3)
(6,4)
(6,5)
(5,6)
(4,7)
(3,8)
(3,9)
(2,10)
```

<u>CONCLUSION :</u>

Thus Bresenham's line drawing algorithm was implemented and the points were plotted satisfying all the 8 test cases.