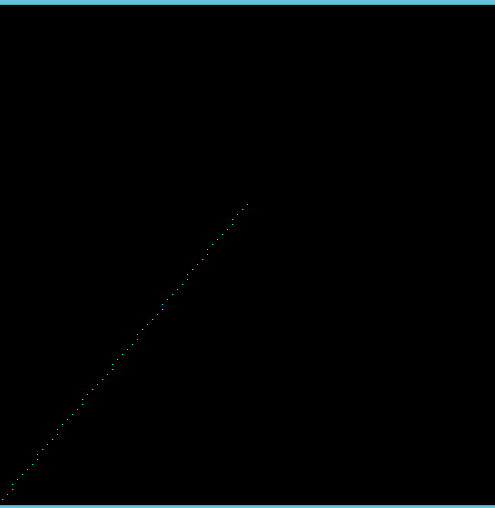
**Screenshot:**

****

**Code:**

#include<stdio.h>

#include<math.h>

#include <GL/gl.h>

#include <GL/glut.h>

void display(void)

{

glClear (GL\_COLOR\_BUFFER\_BIT);

glColor3f (0.0, 1.0, 1.0);

float x = 0;

float y = 0;

float xn = 50.0;

float yn = 60.0;

float xk = x,yk = y;

int k=0;

float dx =(xn -x);

float dy =(yn -y);

float m = dy /dx ;

if(m > 1.0)

{

while(xk <xn && yk <yn)

{

yk = yk + 1;

xk =xk + (1/m);

glBegin(GL\_POINTS);

glVertex2f (roundf(xk), roundf(yk));

printf("\n %f %f",roundf(xk),roundf(yk));

glEnd();

}

glFlush ();

}

else if( m<1.0)

{

while(xk <xn && yk <yn)

{

yk = yk + m;

xk =xk + 1;

glBegin(GL\_POINTS);

glVertex2f (roundf(xk), roundf(yk));

printf("\n %f %f",roundf(xk),roundf(yk));

glEnd();

}

glFlush ();

}

else if( m == 1.0)

{

while(xk <xn && yk <yn)

{

yk = yk + 1;

xk =xk + 1;

glBegin(GL\_POINTS);

glVertex2f (roundf(xk), roundf(yk));

printf("\n %f %f",roundf(xk),roundf(yk));

glEnd();

}

glFlush ();

}

else if( m<-1.0)

{

while(xk <xn && yk <yn)

{

yk = yk + 1;

xk =xk + (1/m);

glBegin(GL\_POINTS);

glVertex2f (roundf(xk), roundf(yk));

printf("\n %f %f",roundf(xk),roundf(yk));

glEnd();

}

glFlush ();

}

else if( -1.0<= m <= 0)

{

while(xk <xn && yk <yn)

{

yk = yk - m;

xk =xk - 1;

glBegin(GL\_POINTS);

glVertex2f (roundf(xk), roundf(yk));

printf("\n %f %f",roundf(xk),roundf(yk));

glEnd();

}

glFlush ();

}

}

void init (void)

{

glClearColor (0.0, 0.0, 0.0, 0.0);

glMatrixMode(GL\_PROJECTION);

glLoadIdentity();

glOrtho(0.0, 100.0, 0.0, 100.0, -1.0, 1.0);

}

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

{

glutInit(&argc, argv);

glutInitDisplayMode (GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize (500, 500);

glutInitWindowPosition (100, 100);

glutCreateWindow ("hello");

init ();

glutDisplayFunc(display);

glutMainLoop();

return 0;

}