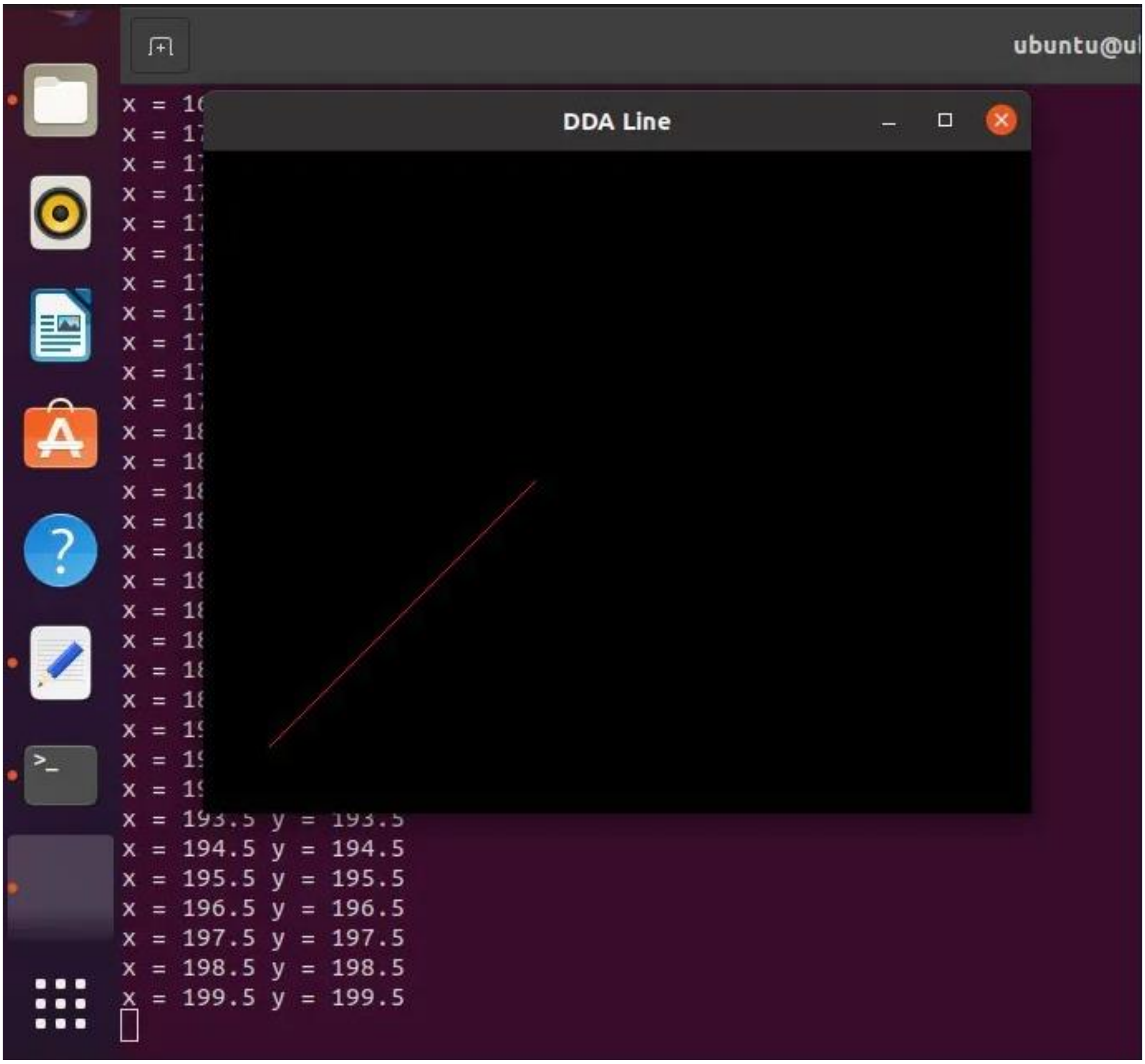
DDA Line Algorithm

```
#include<GL/glut.h>
#include<math.h>
#include<iostream>
using namespace std;


float x_1, x_2, y_1 ,y_2;

int sgn(float a){

    if(a==0){

        return 0;
    }
    if(a<0){

        return -1;
    }
    else
        return 1;


}

void Line(){

    float dy,dx, length;

    dy = y_2 - y_1;
    dx = x_2 - x_1;

    if(abs(dx)>=abs(dy)){

        length = abs(dx);

    }
    else{
        length = abs(dy);
    }

    float xin,yin;

    xin = (x_2-x_1)/length;
    yin = (y_2-y_1)/length;

    float x,y;

    x = x_1 + 0.5 * sgn(xin);
    y = y_1 + 0.5 * sgn(yin);
    int i = 0;
    while(i<=length){
        cout<<"\nx = "<< x <<" y = "<<y;
        glBegin(GL_POINTS);
            glVertex2i(x,y);
        glEnd();
        x = x + xin;
        y = y + yin;
        i++;
    }

    glFlush();
}

void init(void)
{
glClearColor(0,0,0,0);
glColor3f(1.0,0.0,0.0);
gluOrtho2D(0,500,0,400);
glClear(GL_COLOR_BUFFER_BIT);
}

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

    cout<<" Enter x1, y1 point";
    cin>>x_1>>y_1;
    cout<<"\n Enter x2, y2 point";
    cin>>x_2>>y_2;



    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(500,400);
    glutCreateWindow("DDA Line");
    init();
    glutDisplayFunc(Line);
    glutMainLoop();
    return 0;
}
```

OUTPUT:



```
x = 16
x = 17
x = 17
x = 17
x = 17
x = 17
x = 17
x = 17
x = 17
x = 17
x = 17
x = 18
x = 18
x = 18
x = 18
x = 18
x = 18
x = 18
x = 18
x = 18
x = 18
x = 19
x = 19
x = 19
x = 193.5 y = 193.5
x = 194.5 y = 194.5
x = 195.5 y = 195.5
x = 196.5 y = 196.5
x = 197.5 y = 197.5
x = 198.5 y = 198.5
x = 199.5 y = 199.5
```