Bresenham's Line Drawing Algorithm

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

float x_1,y_1,x_2,y_2;

void Line(){

    float dy, dx, m , P;
    dy = y_2 - y_1;
    dx = x_2 - x_1;

    m = dy/dx;

    P = 2*dy - dx;

    int x = x_1, y = y_1;
    if(m<1){

        for(int i=0; i<=dx;i++){

            glBegin(GL_POINTS);
```

```
            glVertex2i(x,y);
        glEnd();

        if(P<0){

            x = x +1;
            y =y;
            P = P + 2*dy;
        }
        else{

            x= x+1;
            y = y+1;
            P = P + 2*dy - 2*dx;

        }
    }
}
else{
    for(int i=0;i<=dy;i++){
        glBegin(GL_POINTS);
            glVertex2i(x,y);
        glEnd();
        if(P<0){
```

```
            x = x;
            y =y+1;
            P = P + 2*dx;
        }
        else{

            x= x+1;
            y = y+1;
            P = P + 2*dx - 2*dy;

        }
    }
  }

  glFlush();
}


void init(){

  glClearColor(1,1,1,0);
  glColor3f(1,0,0);
  gluOrtho2D(0,640,0,400);
  glClear(GL_COLOR_BUFFER_BIT);
}
```

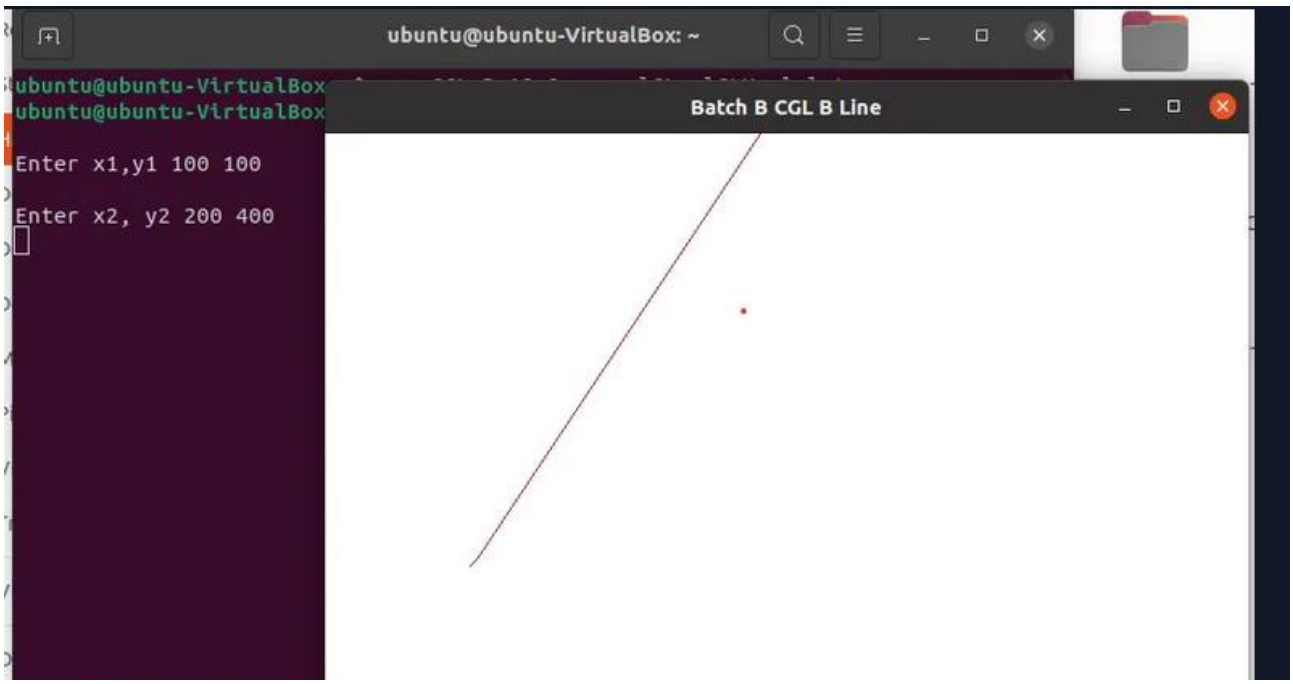
```
int main(int arcg, char** argv){

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

    glutInit(&arcg, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640,400);
    glutCreateWindow("Batch B CGL B Line");
    init();
    glutDisplayFunc(Line);

    glutMainLoop();
}
```

Output:


ubuntu@ubuntu-VirtualBox: ~
ubuntu@ubuntu-VirtualBox
ubuntu@ubuntu-VirtualBox
Enter x1,y1 100 100
Enter x2, y2 200 400
Batch B CGL B Line