Implement Brenham's line drawing algorithm for all types of slope.

```
#include<GL/glut.h>
#include<stdio.h>
int x1, y1, x2, y2;
void draw_pixel(int x, int y)
{
glColor3f(1.0,0.0,0.0); glBegin(GL_POINTS); glVertex2i(x, y);
glEnd();
}
void brenhams_line_draw(int x1, int y1, int x2, int y2)
{
int dx=x2-x1,dy=y2-y1; int p=2*dy*dx;
int twoDy=2*dy;
int twoDyMinusDx=2*(dy-dx); // paranthesis are required int x=x1,y=y1;
if(dx<0)
{
x=x2; y=y2; x2=x1;
}
draw_pixel(x, y); while(x<x2)
{
x++;
if(p<0)
p+=twoDy; else
{
```

```
y++;
p+=twoDyMinusDx;
}
draw_pixel(x, y);
}
}
void myInit()
{
glClearColor(0.0,0.0,0.0,1.0);
glMatrixMode(GL_PROJECTION); glLoadIdentity();
gluOrtho2D(0.0, 500.0, 0.0, 500.0);
glMatrixMode(GL_MODELVIEW);
}
void display()
{
glClear(GL_COLOR_BUFFER_BIT);
brenhams_line_draw(x1, y1, x2, y2); glFlush();
}
void main(int argc, char **argv)
{
printf( "Enter Start Points (x1,y1)\n"); scanf("%d %d", &x1, &y1);
printf( "Enter End Points (x2,y2)\n"); scanf("%d %d", &x2, &y2);

glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); glutInitWindowSize(500, 500);
```

```
glutInitWindowPosition(0, 0);

glutCreateWindow("Bresenham's Line Drawing"); myInit();

glutDisplayFunc(display); glutMainLoop();

}
```

OUTPUT: