

```

#include<stdio.h>
#include<GL/gl.h>
#include<GL/glut.h>
float x,y,x1,z1,x2,y2,dx,dy,step;
void dda()
{
    int xinc,yinc,k;
    glClear(GL_COLOR_BUFFER_BIT);
    glPointSize(5.0);
    glColor3f(0.0,0.5,0.0);
    dx=x2-x1;
    dy=y2-z1;
    if(abs(dx)>abs(dy))
        step=abs(dx);
    else
        step=abs(dy);
    xinc=dx/(float)step;
    yinc=dy/(float)step;
    x=x1;
    y=z1;
    for(k=0;k<=step;k++)
    {
        glBegin(GL_LINES);

        glVertex2f(x,y);
        x=x+xinc;
        y=y+yinc;
        glVertex2f(x,y);
    }
    glEnd();
    glFlush();
}
main(int argc,char **argv)
{
    printf("\nEnter the cordinales of x1 and y1\n");
    scanf("%f%f",&x1,&z1);
    printf("\nEnter the cordinales of x2 and y2\n");
    scanf("%f%f",&x2,&y2);
    glutInit(&argc,argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(500,500);
    glutInitWindowPosition(0,0);
    glutCreateWindow("Line_Drawing");
    glClearColor(0.0,0.0,0.0,0.0);
    glOrtho(-100,100,-100,100,-10,10);
}

```

```

    glutDisplayFunc(dda);
    glutMainLoop();
    return 0;
}

```

Bresenham's line drawing

```

#include<stdio.h>
#include<GL/gl.h>
#include<GL/glut.h>
float x1,x2,z1,y2,x,y,step,p,dx,dy;
void bresenham()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glPointSize(6.0);
    glColor3f(0.2,0.5,0.0);
    int k;
    dx=x2-x1;
    dy=y2-z1;
    step=dx-1;
    p=2*(dy-dx);
    x=x1;
    y=z1;
    for(k=0;k<step;k++)
    {
        glBegin(GL_POINTS);
        glVertex2f(x,y);
        if(p<0)
        {
            x=x+1;
            p=p+(2*dy);
        }
        else
        {
            x=x+1;
            y=y+1;
            p=p+2*(dy-dx);
        }
        glVertex2f(x,y);
    }
    glEnd();
    glFlush();
}

```

Circle drawing

```
#include<stdio.h>
#include<GL/gl.h>
#include<GL/glut.h>
void circle_plotting(float xx,float yy);
float r,xc,yc,x,y,p;
void circle_algorithm()
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(0.5,0.6,0.0);
    glPointSize(2.0);
    x=0;
    y=r;
    p=1-r;
    glBegin(GL_POINTS);
    circle_plotting(x,y);
    while(x<=y)
    {
        if(p<0)
        {
            x=x+1;
            p=p+(2*x)+1;
        }
        else
        {
            x=x+1;
            y=y-1;
            p=p+(2*(x-y))+1;
        }
        circle_plotting(x,y);
    }
    glEnd();
    glFlush();
}

void circle_plotting(float xx, float yy)
{
    glVertex2f(xc+xx,yc+yy);
    glVertex2f(xc-xx,yc-yy);
    glVertex2f(xc-xx,yc+yy);
    glVertex2f(xc+xx,yc-yy);
    glVertex2f(xc+yy,yc+xx);
    glVertex2f(xc-yy,yc+xx);
    glVertex2f(xc+yy,yc-xx);
    glVertex2f(xc-yy,yc-xx);
}
```