

Implement Line Clipping Algorithm : Liang Barsky

```
#include<stdio.h>

#include<graphics.h>

#include<math.h>

#include<dos.h>


int main()

{

int i,gd=DETECT,gm;

int x1,y1,x2,y2,xmin,xmax,ymin,ymax,xx1,xx2,yy1,yy2,dx,dy;

float t1,t2,p[4],q[4],temp;

x1=120;

y1=120;

x2=300;

y2=300;

xmin=100;

ymin=100;

xmax=250;

ymax=250;

initgraph(&gd,&gm,"C://TURBOC3//BGI");

rectangle(xmin,ymin,xmax,ymax);

dx=x2-x1;

dy=y2-y1;

p[0]=-dx;
```

```
p[1]=dx;
p[2]=-dy;
p[3]=dy;
q[0]=x1-xmin;
q[1]=xmax-x1;
q[2]=y1-ymin;
q[3]=ymax-y1;
for(i=0;i<4;i++)
{
    if(p[i]==0)
    {
        printf("line is parallel to one of the clipping boundary");
        if(q[i]>=0)
        {
            if(i<2)
            {
                if(y1<ymin)
                {
                    y1=ymin;
                }
                if(y2>ymax)
                {
                    y2=ymax;
                }
            }
        }
    }
}
```

```
line(x1,y1,x2,y2);
```

```
}
```

```
if(i>1)
```

```
{
```

```
if(x1<xmin)
```

```
{
```

```
x1=xmin;
```

```
}
```

```
if(x2>xmax)
```

```
{
```

```
x2=xmax;
```

```
}
```

```
line(x1,y1,x2,y2);
```

```
}
```

```
}
```

```
}
```

```
}
```

```
t1=0;
```

```
t2=1;
```

```
for(i=0;i<4;i++)
```

```
{
```

```
temp=q[i]/p[i];
```

```
if(p[i]<0)
```

```
{
```

```
if(t1<=temp)
t1=temp;
}
else
{
if(t2>temp)
t2=temp;
}
}
if(t1<t2)
{
xx1 = x1 + t1 * p[1];
xx2 = x1 + t2 * p[1];
yy1 = y1 + t1 * p[3];
yy2 = y1 + t2 * p[3];
line(xx1,yy1,xx2,yy2);
}
delay(5000);
closegraph();
return 0;
}
```

