CG-Assignment Liang Barsky Line Clipping Algorithm

| 11 | 2003088, |
|----------|--|
| Y | C-21, Hriday Keswani , Manha |
| | LIANG BARSKY LINE CLIPPING ALGORITHME 7 |
| | Aim: To implement stalliang Barsky line clipping algorithm. |
| | Algorithm: |
| | the coordinates of line are x1, y1, x2, y2 |
| | Dectarale (xmin, ymin, xmax, ymax); |
| | making p[4] and q[4] |
| | $\rho[0] = (+1)^* (x_2 - x_0)$ |
| | qo[0] = xI - axmin |
| | p[] = (2 - 2) |
| | go [6] = xmare-xt q [1] = xmare-x1 |
| | appellant page |
| | $9 = (-1) * (y, -y_1)$ |
| | p[2] = y1-ymin |
| | $p[3] = y_2 - y_1$ |
| | 9[3] ymax-y1 |
| | (sop(i): |
| | if p[i] = 0, the line is parallel |
| | ifq[i]>0 |
| | if (i<2) |
| | if (yl <ymin)< td=""></ymin)<> |
| | ulcumin |
| | accordingly use set condition to die for conditions their |
| | accordingly we set condition to clip for conditions there for y2 being zyman and following some for x2 and x1; in the case of parallel line. |
| | for 92 heing 2 gman wild journey some get |
| | Ilym the cost of paralle line. |
| | |
| | t1=0 |
| | 12=12 |
| 12/22/20 | |
| | P.T.O. |
| | |
| | |

| 1 | Peoge Na |
|------|--|
| | THE RESIDENCE OF THE STATE OF T |
| | for (iz4): & lie=0 si++ |
| | temp=q[i]/p[i] |
| | H(p[i]<0) |
| | if (t1 <= temp) |
| | tle=temp |
| | else |
| | if (t2 8>temp) |
| | t2=temp & |
| | if Ctl <t2)< th=""></t2)<> |
| | xxl = xl + tl + p[l] |
| | $2x^2 = xl + t^2 p[i]$ |
| | ggl=y1+t1*p[3] |
| | gg2=y1++2*p[3] |
| | line(xx1, yy1)xx2, yy2); |
| | line hase been dipped |
| | else |
| | there was no need to dip the line |
| | Allanda to the fraction |
| | |
| | Usidan Kan and C 21 |
| | Hriday Keswani, C-21, 2003088 |
| 1000 | 2003088 |
| 300 | Control 10 |
| | A Partial was not gibble risitative & took by adjoint response |

Program:

Implementing Liang Barsky line clipping algorithm in C

Code:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main(){
    int gd = DETECT, gm;
    int x1,y1,x2,y2,xwmin,ywmin,xwmax,ywmax,i;
    int xx1,yy1,xx2,yy2,status;
    float t1,t2,p[4],q[4],temp;
    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    printf("Enter the coordinates for line:(xmin = ymin = 100,
xmax = ymax = 250) n";
    scanf("%d%d%d%d",&x1,&y1,&x2,&y2);
    xwmin = ywmin = 100;
    xwmax = ywmax = 250;
    rectangle(xwmin,ywmin,xwmax,ywmax);
    p[0] = (-1)*(x2-x1);
    q[0] = x1-xwmin;
    p[1] = x2-x1;
    q[1] = xwmax - x1;
    p[2] = (-1)*(y2-y1);
    q[2] = y1-ywmin;
    p[3] = y2-y1;
    q[3] = ywmax-y1;
    for(i=0;i<4;i++){
        if(p[i]==0){
     printf("Line is parallel\n");
     if(q[i]>=0){
          if(i<2){
             if(y1<ywmin){</pre>
               y1 = ywmin;
         if(y2>ywmax){
          y2 = ywmax;
          line(x1, y1, x2, y2);
     if(i>1){
         if(x1<xwmin){</pre>
            x1 = xwmin;
         if(x2>xwmax){
            x2 = xwmax;
         }
```

```
}
    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)</pre>
        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);
    printf("Line clipped");
    setcolor(6);
    line(x1,y1,xx1,yy1);
}else{
    printf("Line not clipped");
 }
getch();
clrscr();}
```

Output:

```
Enter the coordinates for line:(xmin = ymin = 100, xmax = ymax = 250)
140
140
270
270
Line clipped

Hriday Keswani
2003008
C-21
```

