**Experiment Name: Implementation of Liang Brasky Line Clipping algorithm.**

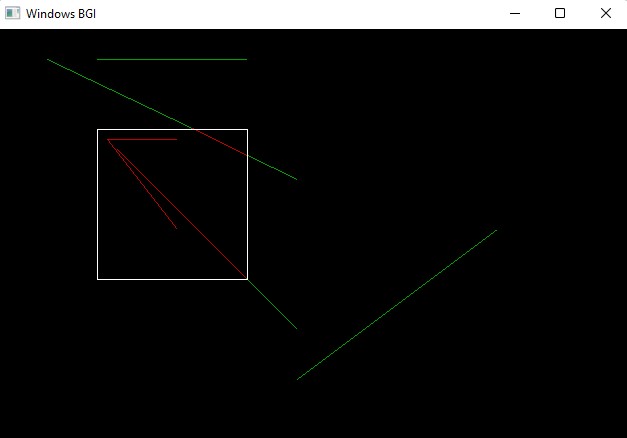
**Introduction:**

Liang Barsky line clipping algorithm uses parametric equations to clip lines in a window. It is more efficient than Sutherland algorithm for line clipping.

**Code in C++:**

| #include<iostream>  #include<graphics.h>  #include<math.h>  #include<dos.h>  using namespace std;  void LiangAlgo(int x1, int y1, int x2, int y2)  {  int i;  Int xmin, xmax, ymin, ymax, xx1, xx2,yy1,yy2,dx,dy;  float t1,t2,p[4],q[4],temp;  xmin=100;  ymin=100;  xmax=250;  ymax=250;  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)  {  if(q[i]>=0)  {  if(i<2)  {  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];  setcolor(2);  line(x1,y1,xx1,yy1);  setcolor(4);  line(xx1,yy1,xx2,yy2);  setcolor(2);  line(xx2,yy2,x2,y2);  }  else  {  setcolor(2);  line(x1,y1,x2,y2);  }  } | {  if(y1<ymin)  {  y1=ymin;  }  if(y2>ymax)  {  y2=ymax;  }  setcolor(2); line(x1,y1,x2,y2);  }  if(i>1)  {  if(x1<xmin)  {  x1=xmin;  }  if(x2>xmax)  {  x2=xmax;  }  setcolor(2); line(x1,y1,x2,y2);  }  }  }  }  t1=0;  t2=1;  for(i=0; i<4; i++)  {  temp=q[i]/p[i];  if(p[i]<0)  int main()  {  int gd=DETECT,gm;  initgraph(&gd,&gm,"c:\\turboc3\\bgi");  int xmin,xmax,ymin,ymax;  xmin=100;  ymin=100;  xmax=250;  ymax=250;  LiangAlgo(120,120,300,300);  LiangAlgo(50,30,300,150);  LiangAlgo(50,30,300,30);  LiangAlgo(300,350,500,200);  LiangAlgo(110,110,180,200);  LiangAlgo(110,110,180,110);  setcolor(15);  rectangle(xmin,ymin,xmax,ymax);  delay(100000);  closegraph();  } |
| --- | --- |

**Output:**



**Discussion:**

The white rectangle is the window and red lines are either clipped or entire lines that fit in the window. The green lines and green parts of clipped lines are outside the window.

—\*---