1. CLIP LINE

```
#include<graphics.h>
#include<iostream.h>
#include<conio.h>

static int LEFT=1,RIGHT=2,BOTTOM=4,TOP=8,xmin,ymin,xmax,ymax;
int getcode(int x,int y)
 {
 int code=0;
 if(y>ymax) code|=TOP;
 if(y<ymin) code|=BOTTOM;
 if(x<xmin) code|=LEFT;
 if(x>xmax) code|=RIGHT;
 return code;
 }
void main()
{
 int gd=DETECT,gm;
 initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
 setcolor(WHITE);
 cout<<"enter windows minimum & maximum values";
 cin>>xmin>>ymin>>xmax>>ymax;
 rectangle(xmin,ymin,xmax,ymax);
 int x1,y1,x2,y2;
 cout<<"enter the endpoints of the line";
 cin>>x1>>y1>>x2>>y2;
 line(x1,y1,x2,y2);
 getch();
 int outcode1=getcode(x1,y1),outcode2=getcode(x2,y2);
 int accept=0;
 while(1)
 {
        float m=(float)(y2-y1)/(x2-x1);
        if(outcode1==0 && outcode2==0)
        {
                accept=1;
                break;
        }
        else if((outcode1 & outcode2)!=0)
        {
                break;
        }
        else
        {
```

```
        int x,y;
        int temp;
        if(outcode1==0)
            temp=outcode2;
        else
            temp=outcode1;
        if(temp & TOP)
        {
            x=x1+(ymax-y1)/m;
            y=ymax;
        }
        else if(temp & BOTTOM)
        {
            x=x1+(ymin-y1)/m;
            y=ymin;
        }
        else if(temp & LEFT)
        {
            x=xmin;
            y=y1+m*(xmin-x1);
        }
        else if(temp & RIGHT)
        {
            x=xmax;
            y=y1+m*(xmax-x1);
        }
        if(temp==outcode1)
        {
            x1=x;
            y1=y;
            outcode1=getcode(x1,y1);
        }
        else
        {
            x2=x;
            y2=y;
            outcode2=getcode(x2,y2);
        }
 }
}

cout<<"after clipping";
if(accept)
 {
    cleardevice();
    rectangle(xmin,ymin,xmax,ymax);
    setcolor(RED);
    line(x1,y1,x2,y2);
        }
  getch();
  closegraph();
}
```