Experiment 13: Program to clip line using midpoint subdivision line clipping algorithm

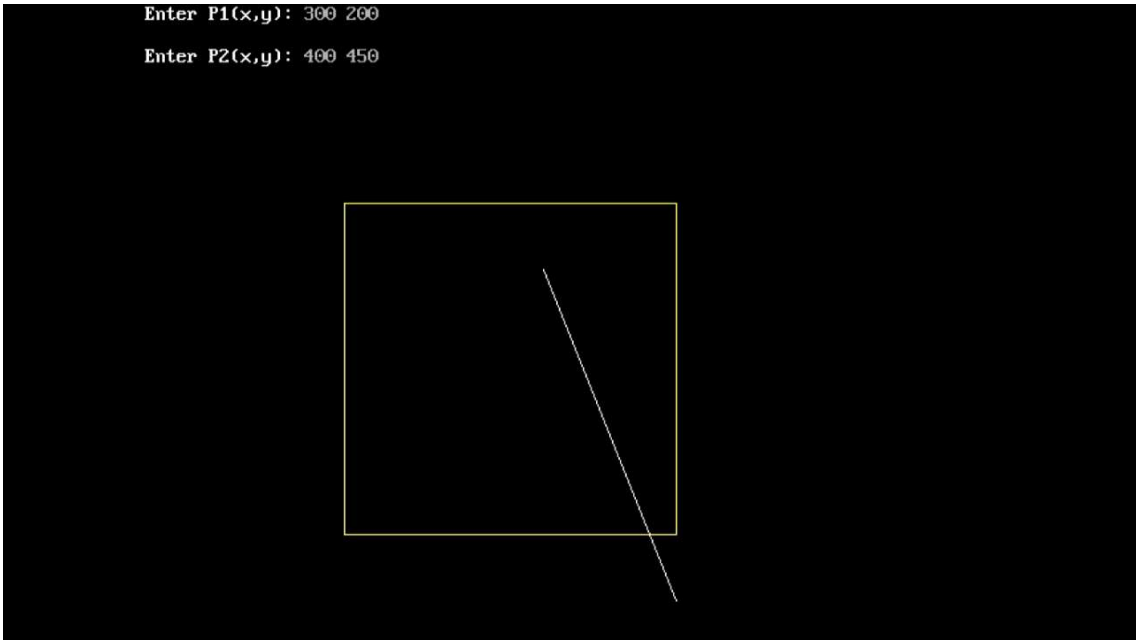
```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
#include<math.h>
#include<stdlib.h>
typedef struct coordinate
{
 int x,y;
 char code[4];
 }PT;

 void drawwindow();
 void drawline (PT p1,PT p2);

 PT setcode(PT p);

 int visible (PT p1,PT p2);

 PT resetendpt (PT p1,PT p2);

 void drawwindow()
{
 setcolor(YELLOW);
 rectangle(150,150,400,400);
}

void main()
{
 int gd=DETECT, gm;
 PT p1,p2,ptemp;
 initgraph(&gd,&gm,"C:\TURBO\BGI");


 setcolor(YELLOW);
 rectangle(150,150,400,400);

   printf("Enter P1(x,y): ");
   scanf("%d%d",&p1.x,&p1.y);

   printf("\nEnter P2(x,y): ");
   scanf("%d%d",&p2.x,&p2.y);

 drawwindow();


 drawline(p1,p2);
```

```
 getch();
 cleardevice();

 drawwindow();
 midsub(p1,p2);

 getch();
 closegraph();
// return(0);

}
 midsub(PT p1,PT p2)
{
 PT mid;
 int ch;

 p1=setcode(p1);
 p2=setcode(p2);
 ch=visible(p1,p2);

 switch(ch)
{
 case 0:
 drawline(p1,p2);
 break;

 case 1:
 break;

 case 2:
 mid.x = p1.x + (p2.x-p1.x)/2;
 mid.y = p1.y + (p2.y-p1.y)/2;

 midsub(p1,mid);

 mid.x = mid.x+1;
 mid.y = mid.y+1;

 midsub(mid,p2);
 break;
}
}

 void drawline (PT p1,PT p2)
{
 setcolor(6);
 line(p1.x,p1.y,p2.x,p2.y);
}
```

```
 PT setcode(PT p)
{
 PT ptemp;
 if(p.y<=100) ptemp.code[0]='1'; else ptemp.code[0]='0';
if(p.y>=400)
 ptemp.code[1]='1';
 else
 ptemp.code[1]='0';
 if (p.x>=450)
 ptemp.code[2]='1';
 else
 ptemp.code[2]='0';
 if (p.x<=150)
 ptemp.code[3]='1';
 else
 ptemp.code[3]='0';
 ptemp.x=p.x;
 ptemp.y=p.y;
 return(ptemp);
}

 int visible (PT p1,PT p2)
{
 int i,flag=0;
 for(i=0;i<4;i++)
{
 if((p1.code[i]!='0')||(p2.code[i]!='0'))
 flag=1;
}
 if(flag==0)
 return(0);
 for(i=0;i<4;i++)
{
 if((p1.code[i]==p2.code[i]) &&(p1.code[i]=='1'))
 flag=0;
}
 if(flag==0)
 return(1);
 return(2);
}
```

Output

Before Clipping:



After Clipping: