

```
// Midpoint Subdivision Algorithm
```

```
#include<stdio.h>
```

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

```
#include<stdlib.h>
```

```
#include<dos.h>
```

```
#include<math.h>
```

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

```
typedef struct coordinate
```

```
{
```

```
int x,y;
```

```
char code[4];
```

```
}PT;
```

```
void drawwindow();
```

```
void drawline (PT p1,PT p2,int cl);
```

```
PT setcode(PT p);
```

```
int visibility (PT p1,PT p2);
```

```
PT resetendpt (PT p1,PT p2);
```

```
main()
```

```
{
```

```
int gd=DETECT, gm,v;
```

```
PT p1,p2,ptemp;
```

```
initgraph(&gd,&gm,"C:\\turbo3\\BGI ");
```

```
cleardevice();
```

```
printf("\n\n\t\tENTER END-POINT 1 (x,y): ");
```

```
scanf("%d%d",&p1.x,&p1.y);
```

```
printf("\n\n\t\tENTER END-POINT 2 (x,y): ");
```

```
scanf("%d%d",&p2.x,&p2.y);
```

```
cleardevice();
```

```
drawwindow();  
getch();  
drawline(p1,p2,15);  
getch();
```

```
cleardevice();  
drawwindow();  
midsub(p1,p2);  
getch();  
closegraph();  
return(0);  
}
```

```
midsub(PT p1,PT p2)  
{  
    PT mid;  
    int v;  
    p1=setcode(p1);  
    p2=setcode(p2);  
    v=visibility(p1,p2);  
    switch(v)  
    {  
        case 0: /* Line completely visible */  
            drawline(p1,p2,15);  
            break;  
        case 1: /* Line completely invisible */  
            break;  
        case 2: /* line partly visible */  
            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 drawwindow()
{
setcolor(RED);
line(150,100,450,100);
line(450,100,450,400);
line(450,400,150,400);
line(150,400,150,100);
}

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

PT setcode(PT p)
{
PT ptemp;
if(p.y<=100)
ptemp.code[0]='1'; /* TOP */
else
ptemp.code[0]='0';
if(p.y>=400)
ptemp.code[1]='1'; /* BOTTOM */
else

```

```

ptemp.code[1]='0';
if (p.x>=450)
ptemp.code[2]='1'; /* RIGHT */
else
ptemp.code[2]='0';
if (p.x<=150) /* LEFT */
ptemp.code[3]='1';
else
ptemp.code[3]='0';
ptemp.x=p.x;
ptemp.y=p.y;
return(ptemp);
}
int visibility (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);  
}
```