1. CLIP POLYGON

```
#include<iostream.h>
#include<graphics.h>
#include<conio.h>
#define round(a)((int)(a+0.5))
int k;
float xmin,ymin,xmax,ymax,arr[20],m;
void clipleft(float x1,float y1,float x2,float y2)
{
        if(x2-x1)
        m=(y2-y1)/(x2-x1);
        else
        m=10000;
        if(x1>=xmin && x2>=xmin)
        {
                arr[k]=x2;
                arr[k+1]=y2;
                k+=2;
        }
        if(x1<xmin && x2>=xmin)
        {
                arr[k]=xmin;
                arr[k+1]=y1+m*(xmin-x1);
                arr[k+2]=x2;
                arr[k+3]=y2;
                k+=4;
        }
        if(x1>=xmin && x2<xmin)
        {
                arr[k]=xmin;
                arr[k+1]=y1+m*(xmin-x1);
                k+=2;
        }
}

void cliptop(float x1,float y1,float x2,float y2)
{
        if(y2-y1)
        m=(x2-x1)/(y2-y1);
        else
        m=10000;
        if(y1>=ymax && y2>=xmax)
        {
```

```
                arr[k]=x2;
                arr[k+1]=y2;
                k+=2;
        }
        if(y1<ymax && y2>=ymax)
        {
                arr[k]=x1+m*(ymax-y1);
                arr[k+1]=ymax;
                arr[k+2]=x2;
                arr[k+3]=y2;
                k+=4;
        }
        if(y1<=ymax && y2<ymax)
        {
                arr[k]=x1+m*(ymax-y1);
                arr[k+1]=ymax;
                k+=2;
        }
}

void clipright(float x1,float y1,float x2,float y2)
{
        if(x2-x1)
        m=(y2-y1)/(x2-x1);
        else
        m=10000;
        if(x1<=xmax && x2<=xmax)
        {
                arr[k]=x2;
                arr[k+1]=y2;
                k+=2;
        }
        if(x1>xmax && x2<=xmax)
        {
                arr[k]=xmax;
                arr[k+1]=y1+m*(xmax-x1);
                arr[k+2]=x2;
                arr[k+3]=y2;
                k+=4;
        }
        if(x1<xmax && x2>xmax)
        {
                arr[k]=xmax;
                arr[k+1]=y1+m*(xmax-x1);
```

```
                k+=2;
        }
}

void clipbottom(float x1,float y1,float x2,float y2)
{
        if(y2-y1)
        m=(x2-x1)/(y2-y1);
        else
        m=10000;
        if(y1>=ymin && y2>=xmin)
        {
                arr[k]=x2;
                arr[k+1]=y2;
                k+=2;
        }
        if(y1<ymin && y2>=ymin)
        {
                arr[k]=x1+m*(ymin-y1);
                arr[k+1]=ymin;
                arr[k+2]=x2;
                arr[k+3]=y2;
                k+=4;
        }
        if(y1<=ymin && y2<ymin)
        {
                arr[k]=x1+m*(ymin-y1);
                arr[k+1]=ymin;
                k+=2;
        }
}

void main()
{
        int gd=DETECT,gm,n,poly[20],i;
        float x0,y0,x1,y1,polyy[20];
        clrscr();
        cout<<"coordinates for clipping window";
        cin>>xmin>>ymin>>xmax>>ymax;
        cout<<"Polygon 'N'";
        cin>>n;
        cout<<"enter the coordinates:";
        for(i=0;i<2*n;i++)
        cin>>polyy[i];
```

```
polyy[i]=polyy[0];
polyy[i+1]=polyy[1];
for(i=0;i<2*n+2;i++)
poly[i]=round(polyy[i]);
initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
setcolor(RED);
rectangle(xmin,ymin,xmax,ymax);
cout<<"Unclipped polygon";
setcolor(WHITE);
fillpoly(n,poly);
getch();
cleardevice();

k=0;
for(i=0;i<2*n;i+=2)
clipleft(polyy[i],polyy[i+1],polyy[i+2],polyy[i+3]);
n=k/2;
for(i=0;i<k;i++)
        polyy[i]=arr[i];
polyy[i]=polyy[0];
polyy[i+1]=polyy[1];
k=0;

for(i=0;i<2*n;i+=2)
cliptop(polyy[i],polyy[i+1],polyy[i+2],polyy[i+3]);
n=k/2;
for(i=0;i<k;i++)
        polyy[i]=arr[i];
polyy[i]=polyy[0];
polyy[i+1]=polyy[1];
k=0;

for(i=0;i<2*n;i+=2)
clipright(polyy[i],polyy[i+1],polyy[i+2],polyy[i+3]);
n=k/2;
for(i=0;i<k;i++)
        polyy[i]=arr[i];
polyy[i]=polyy[0];
polyy[i+1]=polyy[1];
k=0;

for(i=0;i<2*n;i+=2)
clipbottom(polyy[i],polyy[i+1],polyy[i+2],polyy[i+3]);
for(i=0;i<k;i++)
```

```
            poly[i]=round(arr[i]);
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
        cout<<"clipped polygon";
        setcolor(RED);
        rectangle(xmin,ymin,xmax,ymax);
            setcolor(WHITE);
        if(k)
            drawpoly(k/2,poly);
        getch();
        closegraph();
}
```