// Name – Dahiwal Satyam Santoshkumar

// Roll Number – 207019

#include <iostream>

using namespace std;

#include <math.h>

#include <GL/glut.h>

typedef struct

{

float x;

float y;

}PT;

int n,i,j;

void MyMouse(int ,int ,int ,int );

void drawpolygon();

void left(PT,PT[],PT[]);

void right(PT,PT[],PT[]);

void top(PT,PT[],PT[]);

void bottom(PT,PT[],PT[]);

void MyInit();

void Display();

PT d,p1,p2,p[20],pi1,pi2,pp[20];

int main(int argc, char \*\*argv)

{

cout<<"\nENTER COORDINATES FOR WINDOW";

cout<<"\nENTER ( Xmin , Ymin ):";

cin>>p1.x>>p1.y;

cout<<"\nENTER ( Xmax , Ymax):";

cin>>p2.x>>p2.y;

cout<<"\n ENTER NUMBER OF VERTEX : ";

cin>>n;

for(i=0;i<n;i++)

{

cout<<"\n ENTER COORDINATES OF VERTEX : ";cout<<i+1;

cout<<"\n ENTER ( X,Y ) : ";

cin>>p[i].x>>p[i].y;

}

p[i].x=p[0].x;

p[i].y=p[0].y;

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowPosition(0,0);

glutInitWindowSize(650,650);

glutCreateWindow("POLYGON Clipping");

MyInit();

glutDisplayFunc(Display);

glutMouseFunc(MyMouse);

glutMainLoop();

return 0;

}

void left() //LEFT BOUNDARY CLIPPING ALGO

{

i=0;

j=0;

for(i=0;i<n;i++)

{

if(p[i].x<p1.x && p[i+1].x>=p1.x)

{

if(p[i+1].x-p[i].x!=0)

{

pp[j].y=(p[i+1].y-p[i].y)/(p[i+1].x-p[i].x)\*(p1.x-p[i].x)+p[i].y;

}

else

{

pp[j].y=p[i].y;

}

pp[j].x=p1.x;

j++;

pp[j].x=p[i+1].x;

pp[j].y=p[i+1].y;

j++;

}

if(p[i].x>p1.x && p[i+1].x>=p1.x)

{

pp[j].y=p[i+1].y;

pp[j].x=p[i+1].x;

j++;

}

if(p[i].x>p1.x && p[i+1].x<=p1.x)

{

if(p[i+1].x-p[i].x!=0)

{

pp[j].y=(p[i+1].y-p[i].y)/(p[i+1].x-p[i].x)\*(p1.x-p[i].x)+p[i].y;

}

else

{

pp[j].y=p[i].y;

}

pp[j].x=p1.x;

j++;

}

}

for(i=0;i<j;i++)

{

p[i].x=pp[i].x;

p[i].y=pp[i].y;

}

p[i].x=pp[0].x;

p[i].y=pp[0].y;

n=j;

}

void right() //RIGHT BOUNDARY CLIPPING ALGO

{

i=0;

j=0;

for(i=0;i<n;i++)

{

if(p[i].x>p2.x&&p[i+1].x<=p2.x)

{

if(p[i+1].x-p[i].x!=0)

{

pp[j].y=(p[i+1].y-p[i].y)/(p[i+1].x-p[i].x)\*(p2.x-p[i].x)+p[i].y;

}

else

{

pp[j].y=p[i].y;

}

pp[j].x=p2.x;

j++;

pp[j].x=p[i+1].x;

pp[j].y=p[i+1].y;

j++;

}

if(p[i].x<p2.x&&p[i+1].x<=p2.x)

{

pp[j].y=p[i+1].y;

pp[j].x=p[i+1].x;

j++;

}

if(p[i].x<p2.x&&p[i+1].x>=p2.x)

{

if(p[i+1].x-p[i].x!=0)

{

pp[j].y=(p[i+1].y-p[i].y)/(p[i+1].x-p[i].x)\*(p2.x-p[i].x)+p[i].y;

}

else

{

pp[j].y=p[i].y;

}

pp[j].x=p2.x;

j++;

}

}

for(i=0;i<j;i++)

{

p[i].x=pp[i].x;

p[i].y=pp[i].y;

}

p[i].x=pp[0].x;

p[i].y=pp[0].y;

n=j;

}

void bottom() //BOTTOM BOUNDARY CLIPPING ALGO

{

i=0;

j=0;

for(i=0;i<n;i++)

{

if(p[i].y>p2.y&&p[i+1].y<=p2.y)

{

if(p[i+1].y-p[i].y!=0)

{

pp[j].x=(p[i+1].x-p[i].x)/(p[i+1].y-p[i].y)\*(p2.y-p[i].y)+p[i].x;

}

else

{

pp[j].x=p[i].x;

}

pp[j].y=p2.y;

j++;

pp[j].x=p[i+1].x;

pp[j].y=p[i+1].y;

j++;

}

if(p[i].y<p2.y&&p[i+1].y<=p2.y)

{

pp[j].y=p[i+1].y;

pp[j].x=p[i+1].x;

j++;

}

if(p[i].y<p2.y&&p[i+1].y>=p2.y)

{

if(p[i+1].y-p[i].y!=0)

{

pp[j].x=(p[i+1].x-p[i].x)/(p[i+1].y-p[i].y)\*(p2.y-p[i].y)+p[i].x;

}

else

{

pp[j].x=p[i].x;

}

pp[j].y=p2.y;

j++;

}

}

for(i=0;i<j;i++)

{

p[i].x=pp[i].x;

p[i].y=pp[i].y;

}

p[i].x=pp[0].x;

p[i].y=pp[0].y;

n=j;

}

void top() //ABOVE BOUNDARY CLIPPING ALGO

{

i=0;

j=0;

for(i=0;i<n;i++)

{

if(p[i].y<p1.y&&p[i+1].y>=p1.y)

{

if(p[i+1].y-p[i].y!=0)

{

pp[j].x=(p[i+1].x-p[i].x)/(p[i+1].y-p[i].y)\*(p1.y-p[i].y)+p[i].x;

}

else

{

pp[j].x=p[i].x;

}

pp[j].y=p1.y;

j++;

pp[j].x=p[i+1].x;

pp[j].y=p[i+1].y;

j++;

}

if(p[i].y>p1.y&&p[i+1].y>=p1.y)

{

pp[j].y=p[i+1].y;

pp[j].x=p[i+1].x;

j++;

}

if(p[i].y>p1.y&&p[i+1].y<=p1.y)

{

if(p[i+1].y-p[i].y!=0)

{

pp[j].x=(p[i+1].x-p[i].x)/(p[i+1].y-p[i].y)\*(p1.y-p[i].y)+p[i].x;

}

else

{

pp[j].x=p[i].x;

}

pp[j].y=p1.y;

j++;

}

}

for(i=0;i<j;i++)

{

p[i].x=pp[i].x;

p[i].y=pp[i].y;

}

p[i].x=pp[0].x;

p[i].y=pp[0].y;

n=j;

}

void drawpolygon() //FOR CONSTRUCTING POLYGON

{

for(i=0;i<n-1;i++)

{

glBegin(GL\_LINES);

glColor3f(0.0,0.0,1.0);

glVertex2d(p[i].x,p[i].y);

glVertex2d(p[i+1].x,p[i+1].y);

glEnd();

}

glBegin(GL\_LINES);

glVertex2d(p[i].x,p[i].y);

glVertex2d(p[0].x,p[0].y);

glEnd();

}

void MyMouse(int button,int state,int x,int y) //MOUSE INTERFACING

{

if(button==GLUT\_RIGHT\_BUTTON && state==GLUT\_DOWN)

{

//ON CLICK OF RIGHT BUTTON OF MOUSE POLYGON WILL BE CLIPPED

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glBegin(GL\_LINE\_LOOP);

glColor3f(0.0,1.0,0.0);

glVertex2f(p1.x,p1.y);

glVertex2f(p2.x,p1.y);

glVertex2f(p2.x,p2.y);

glVertex2f(p1.x,p2.y);

glEnd();

left();

right();

top();

bottom();

drawpolygon();

}

glFlush();

}

void MyInit()

{

glClearColor(1.0,1.0,1.0,1);

glColor3f(0.0,1.0,0.0);

glPointSize(5.0);

gluOrtho2D(0,640,0,480);

}

void Display()

{

float x,y,dx,dy,length;

glClear(GL\_COLOR\_BUFFER\_BIT | GL\_DEPTH\_BUFFER\_BIT);

glBegin(GL\_LINE\_LOOP);

glColor3f(0.0,1.0,0.0);

glVertex2f(p1.x,p1.y);

glVertex2f(p2.x,p1.y);

glVertex2f(p2.x,p2.y);

glVertex2f(p1.x,p2.y);

glEnd();

drawpolygon();

glFlush();

}

Output

satyam@ubuntu:~$ g++ six.cpp -lglut -lGL -lGLEW -lGLU -o six

satyam@ubuntu:~$ ./six

ENTER COORDINATES FOR WINDOW

ENTER ( Xmin , Ymin ):40

40

ENTER ( Xmax , Ymax):400

400

ENTER NUMBER OF VERTEX : 3

ENTER COORDINATES OF VERTEX : 1

ENTER ( X,Y ) : 20

20

ENTER COORDINATES OF VERTEX : 2

ENTER ( X,Y ) : 70

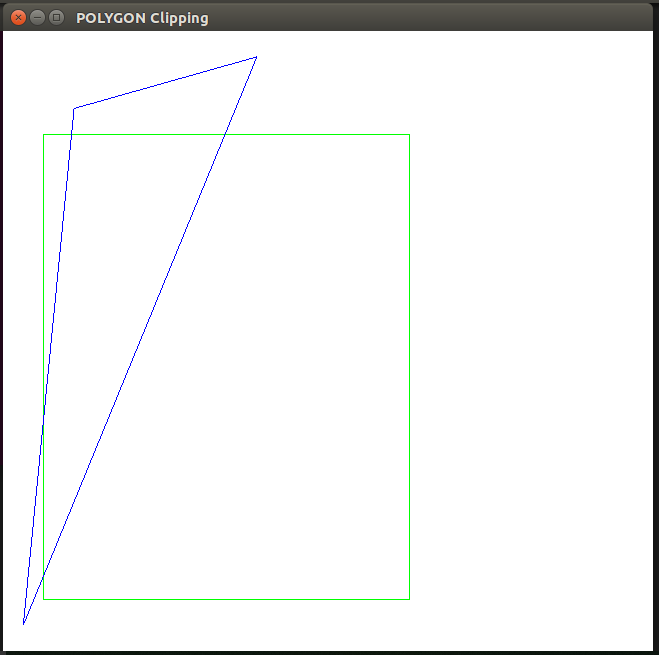
420

ENTER COORDINATES OF VERTEX : 3

ENTER ( X,Y ) : 250

460

Before Clipping:



After Clipping:

