#include<stdio.h>

#include<GL/glut.h> // Header File For The GLUT Library

#include<stdlib.h>

#include<math.h>

//#include<conio.h>

int v,wxmin,wymin,wxmax,wymax,p[10][2];

void init(void)

{

glClearColor(0.0,0.0,0.0,0.0);

gluOrtho2D(-600.0,600.0,-600.0,600.0);

}

accept\_poly(int p[10][2])

{

int i,v;

printf("\nEnter no. of vertices in a polygon:");

scanf("%d",&v);

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

{

printf("\n Enter the %d co-ordinate(x%d,y%d): -->",i+1,i+1,i+1);

scanf("%d%d",&p[i][0],&p[i][1]);

}

return(v);

}

void draw\_poly(int p[10][2],int v)

{

int i;

glBegin(GL\_LINES);

glVertex2i(wxmin,wymin);

glVertex2i(wxmin,wymax);

glVertex2i(wxmin,wymax);

glVertex2i(wxmax,wymax);

glVertex2i(wxmax,wymax);

glVertex2i(wxmax,wymin);

glVertex2i(wxmax,wymin);

glVertex2i(wxmin,wymin);

//draw\_poly(p,v);

glEnd();

//glutMainLoop();

glFlush();

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

{

glColor3f(0.0,1.0,0.0);

glBegin(GL\_LINES);

glVertex2i(p[i][0],p[i][1]);

glVertex2i(p[(i+1)%v][0],p[(i+1)%v][1]);

glEnd();

glFlush();

}

}

int leftclip(int p[10][2],int v,int wxmin)

{

int i,t[10][2];

int k=0;

int x1,y1,x2,y2;

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

{

if(p[i][0] >= wxmin)

{

t[k][0]=p[i][0];

t[k][1]=p[i][1];

k++;

}

if((p[i][0] > wxmin && p[(i+1)%v][0] < wxmin) || (p[i][0] < wxmin && p[(i+1)%v][0] > wxmin))

{

x1=p[i][0];

y1=p[i][1];

x2=p[(i+1)%v][0];

y2=p[(i+1)%v][1];

t[k][1] = y1 + (wxmin-x1) \* ((float)(y2-y1)/(x2-x1));

t[k][0] = wxmin;

k++;

}

}

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

{

p[i][0] = t[i][0];

p[i][1] = t[i][1];

}

// draw\_poly(p,v);

return(k);

}

int rightclip(int p[10][2],int v,int wxmax)

{

int i,t[10][2];

int k=0;

int x1,y1,x2,y2;

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

{

if(p[i][0] <= wxmax)

{

t[k][0]=p[i][0];

t[k][1]=p[i][1];

k++;

}

if((p[i][0] > wxmax && p[(i+1)%v][0] < wxmax) || (p[i][0] < wxmax && p[(i+1)%v][0] > wxmax))

{

x1=p[i][0];

y1=p[i][1];

x2=p[(i+1)%v][0];

y2=p[(i+1)%v][1];

t[k][1] = y1 + (wxmax-x1) \* ((float)(y2-y1)/(x2-x1));

t[k][0] = wxmax;

k++;

}

}

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

{

p[i][0] = t[i][0];

p[i][1] = t[i][1];

}

return(k);

}

int bottomclip(int p[10][2],int v,int wymin)

{

int i,t[10][2];

int k=0;

int x1,y1,x2,y2;

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

{

if(p[i][1] >= wymin)

{

t[k][0]=p[i][0];

t[k][1]=p[i][1];

k++;

}

if((p[i][1] > wymin && p[(i+1)%v][1] < wymin) || (p[i][1] < wymin && p[(i+1)%v][1] > wymin))

{

x1=p[i][0];

y1=p[i][1];

x2=p[(i+1)%v][0];

y2=p[(i+1)%v][1];

t[k][0] = x1 + (wymin-y1) \* ((float)(x2-x1)/(y2-y1));

t[k][1] = wymin;

k++;

}

}

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

{

p[i][0] = t[i][0];

p[i][1] = t[i][1];

}

return(k);

}

int topclip(int p[10][2],int v,int wymax)

{

int i,t[10][2];

int k=0;

int x1,y1,x2,y2;

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

{

if(p[i][1] <= wymax)

{

t[k][0]=p[i][0];

t[k][1]=p[i][1];

k++;

}

if((p[i][1] > wymax && p[(i+1)%v][1] < wymax) || (p[i][1] < wymax && p[(i+1)%v][1] > wymax))

{

x1=p[i][0];

y1=p[i][1];

x2=p[(i+1)%v][0];

y2=p[(i+1)%v][1];

t[k][0] = x1 + (wymax-y1) \* ((float)(x2-x1)/(y2-y1));

t[k][1] = wymax;

k++;

}

}

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

{

p[i][0] = t[i][0];

p[i][1] = t[i][1];

}

return(k);

}

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

{

/\* request auto detection \*/

// int gd = DETECT, gm, errorcode;

printf("\nEnter the window co-ordinates:");

printf("\nWxmin ==> ");

scanf("%d",&wxmin);

printf("\nWymin ==> ");

scanf("%d",&wymin);

printf("\nWxmax ==> ");

scanf("%d",&wxmax);

printf("\nWymax ==> ");

scanf("%d",&wymax);

v=accept\_poly(p);

glutInit(&argc,argv);

glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

glutInitWindowSize(800,800); //window size is imp...

glutInitWindowPosition(100,100);

glutCreateWindow("translation");

init(); /\* Initialize window. \*/

glClear(GL\_COLOR\_BUFFER\_BIT);

draw\_poly(p,v);

int inp;

scanf("%d", &inp);

v=leftclip(p,v,wxmin);

draw\_poly(p,v);

v=rightclip(p,v,wxmax);

// getch();

v=bottomclip(p,v,wymin);

// getch();

// return 0;

v=topclip(p,v,wymax);

//getch();

glClear(GL\_COLOR\_BUFFER\_BIT);

draw\_poly(p,v);

glFlush();

glutMainLoop();

/\* clean up \*/

return 0;

}