#include <iostream>

#include <cmath>

#include <graphics.h>

#include "G:\Computer Graphics\00\_Basic\_tools\Algorithms.h"

using namespace std;

// Cohen-Sutherland line clipping algorithm

// TDRL - Top Down Right Left

int g\_code(float x\_min, float x\_max, float y\_min, float y\_max, float x, float y){

int code = 0;

if (y > y\_max)

code |= 8;

if (y < y\_min)

code |= 4;

if (x > x\_max)

code |= 2;

if (x < x\_min)

code |= 1;

return code;

}

void get\_xy(float &x, float &y, int code, float m, float c, float x\_max, float x\_min, float y\_max, float y\_min){

// Compares x and x\_max and x\_min

// If x is out of range, x = x\_max or x\_min and y is derived accordingly

// y is returned

if ((code&8)==8){

cout<<"Top"<<endl;

y = y\_max;

x = (y\_max - c)/m;

return;

}

if((code&4)==4){

cout<<"Down"<<endl;

y = y\_min;

x = (y\_min - c)/m;

return;

}

if((code&2)==2){

cout<<"Right"<<endl;

x = x\_max;

y = m\*x\_max + c;

return;

}

if((code&1)==1){

cout<<"Left"<<endl;

x = x\_min;

y = m\*x\_min + c;

return;

}

}

void clipping(float x\_min, float x\_max, float y\_min, float y\_max, float x1, float y1, float x2, float y2){

int code1, code2;

code1 = g\_code(x\_min, x\_max, y\_min, y\_max, x1, y1);

code2 = g\_code(x\_min, x\_max, y\_min, y\_max, x2, y2);

float xn1=x1, xn2=x2, yn1=y1, yn2=y2;

float m = (y2-y1)/(x2-x1);

float c = y1-(m\*x1);

if((code1 | code2) == 0){

q\_line(x1, y1, x2, y2);

}

else if((code1 & code2)!=0){

return;

}

else{

// when code1 is in and code2 is out

// when code1 is out and code2 is in

// when both code1 and code2 are out

get\_xy(xn1, yn1, code1, m, c, x\_max, x\_min, y\_max, y\_min);

cout<<xn1<<" "<<yn1<<endl;

get\_xy(xn2, yn2, code2, m, c, x\_max, x\_min, y\_max, y\_min);

cout<<xn2<<" "<<yn2<<endl;

// cout<<xn1<<" "<<yn1<<" "<<xn2<<" "<<yn2<<endl;

q\_line(xn1, yn1, xn2, yn2, 9);

}

}

int main(){

initwindow(1200,800,"Graphics");

q\_line(300,300,600,300);

q\_line(600,300,600,600);

q\_line(600,600,300,600);

q\_line(300,600,300,300);

q\_line(200,400,400,250);

// q\_line(100,700,400,350);

clipping(300,600,300,600,200,400,400,250);

// clipping(300,600,300,600,100,700,400,350);

getch();

return 0;

}