```
#include<GL/glut.h>
#define outcode int
double xmin = 100, ymin = 100, xmax = 200, ymax = 200; // clipping window
double xvmin=300, yvmin=300,xvmax=400, yvmax=400; // view port
double x0, y0, x1, y1;
const int RIGHT=8;
const int LEFT=2;
const int TOP=4;
const int BOTTOM=1;
outcode ComputeOutCode(double x, double y);
void CohenSutherland(double x0, double y0, double x1, double y1)
{
outcode outcode0, outcode1, outcodeOut;
bool accept=false, done=false;
outcode0=ComputeOutCode(x0,y0);
outcode1=ComputeOutCode(x1,y1);
do
{
if(!(outcode0|outcode1))
{
accept=true; // Line is completely visible
done=true;
}
else if(outcode0&outcode1)
done=true; // Line is completely invisible
else // Line is partially visible
{
double x,y;
outcodeOut=outcode0?outcode0:outcode1;
if(outcodeOut&TOP)
{
x=x0+(x1-x0)*(ymax-y0)/(y1-y0);
y=ymax;
}
else if(outcodeOut&BOTTOM)
{
x=x0+(x1-x0)*(ymin-y0)/(y1-y0);
y=ymin;
}
else if(outcodeOut&RIGHT)
{
y=y0+(y1-y0)*(xmax-x0)/(x1-x0);
x=xmax;
}
else
{
y=y0+(y1-y0)*(xmin-x0)/(x1-x0);
x=xmin;
}
if(outcodeOut==outcode0)
```

```
{
x0=x;
y0=y;
outcode0=ComputeOutCode(x0,y0);
}
else
{
x1=x;
y1=y;
outcode1=ComputeOutCode(x1,y1);
}
}
}while(!done);
if(accept)
{
double sx=(xvmax-xvmin)/(xmax-xmin);
double sy=(yvmax-yvmin)/(ymax-ymin);
double vx0=xvmin+(x0-xmin)*sx;
double vy0=yvmin+(y0-ymin)*sy;
double vx1=xvmin+(x1-xmin)*sx;
double vy1=yvmin+(y1-ymin)*sy;
glColor3f(1.0, 1.0, 1.0);
glBegin(GL_LINE_LOOP);
glVertex2f(xvmin, yvmin);
glVertex2f(xvmax, yvmin);
glVertex2f(xvmax, yvmax);
glVertex2f(xvmin, yvmax);
glEnd();
glColor3f(1.0,1.0,0.0);
glBegin(GL_LINES);
glVertex2d(vx0, vy0);
glVertex2d(vx1,vy1);
glEnd();
}
}
outcode ComputeOutCode(double x, double y)
{
outcode code=0; // Assign region code
if(y>ymax)
code=TOP;
else if(y<ymin)
code=BOTTOM;
if(x>xmax)
code=RIGHT;
else if(x<xmin)
code=LEFT;
return code;
}
void display()
{
```

```
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1.0,1.0,0.0);
glBegin(GL_LINE_LOOP); // Draw clipping window
glVertex2f(xmin, ymin);
glVertex2f(xmax, ymin);
glVertex2f(xmax, ymax);
glVertex2f(xmin, ymax);
glEnd();
glFlush();
}
void myinit()
{
glClearColor(0.0,0.0,0.0,1.0);
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluOrtho2D(0.0,500.0,0.0,500.0);
}
void myKeyboard(unsigned char key, int mouseX, int mouseY)
{
switch (key)
{
 case 27: // Press ESC key to exit
 exit(0);
}
}
void myMouse(int button, int state, int x, int y)
{
static int pt = 0;
if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN)
{
glColor3f(1.0, 1.0,1.0);
if (pt == 0) // Get the start point of the line
{
 x0 = x;
 y0= 500-y;
 pt++;
}
else if (pt == 1) // Get the end point of the line
{
 x1 = x;
 y1 = 500-y;
 glBegin(GL_LINES); // Draw initial line
 glVertex2f(x0,y0);
 glVertex2f(x1,y1);
 glEnd();
 pt = 0;
}
}
CohenSutherland(x0,y0,x1,y1);
glFlush(); // Send output to display
```

```
}
int main(int argc, char** argv)
{
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
glutInitWindowSize(500,500);
glutInitWindowPosition(0,0);
glutCreateWindow("Cohen-Sutherland Line Clipping");
glutMouseFunc(myMouse); // Register mouse function
glutKeyboardFunc(myKeyboard); // Register keyboard function
glutDisplayFunc(display);
myinit();
glutMainLoop();
return 0;
}
```