

```

#include <stdio.h>
#include <stdlib.h>
#include <graphics.h>
#include <conio.h>

#define MAX 20
#define TRUE 1
#define FALSE 0

int top=1, bottom=2, right=4, left=8;
typedef unsigned int outcode;

outcode compute_outcode(int x, int y, int xmin, int ymin, int xmax, int ymax)
{
    outcode oc = 0;

    if (y > ymax)
        oc |= top;
    else if (y < ymin)
        oc |= bottom;

    if (x > xmax)
        oc |= right;
    else if (x < xmin)
        oc |= left;

    return oc;
}

void cohen_sutherland (double x1, double y1, double x2, double y2, double xmin, double ymin, double xmax, double
ymax)
{
    int accept;
    int done;
    outcode outcode1, outcode2;

    accept = FALSE;
    done = FALSE;

    outcode1 = compute_outcode (x1, y1, xmin, ymin, xmax, ymax);
    outcode2 = compute_outcode (x2, y2, xmin, ymin, xmax, ymax);

    do
    {
        if (outcode1 == 0 && outcode2 == 0)
        {
            accept = TRUE;
            done = TRUE;
        }
        else if (outcode1 & outcode2)
            done = TRUE;
        else
        {
            double x, y;
            int outcode_ex = outcode1 ? outcode1 : outcode2;

            if (outcode_ex & top)
            {
                x = x1 + (x2 - x1) * (ymax - y1) / (y2 - y1);
                y = ymax;
            }
            else if (outcode_ex & bottom)
            {
                x = x1 + (x2 - x1) * (ymin - y1) / (y2 - y1);
                y = ymin;
            }
            else if (outcode_ex & right)
            {
                y = y1 + (y2 - y1) * (xmax - x1) / (x2 - x1);
                x = xmax;
            }
            else
            {
                y = y1 + (y2 - y1) * (xmin - x1) / (x2 - x1);

```

```

        x = xmin;
    }
    if (outcode_ex == outcode1)
    {
        x1 = x;
        y1 = y;
        outcode1 = compute_outcode (x1, y1, xmin, ymin, xmax, ymax);
    }
    else
    {
        x2 = x;
        y2 = y;
        outcode2 = compute_outcode (x2, y2, xmin, ymin, xmax, ymax);
    }
}
} while (done == FALSE);

if (accept == TRUE)
    line (x1, y1, x2, y2);
}

int main()
{
    int n, i, j, ln[MAX][4], clip[4];
    printf("Enter universal coordinates:");

    for (i=0; i<4; i++)
    {
        if(i==0)
            printf("\nXmin = ");
        if(i==1)
            printf("Ymin = ");
        if(i==2)
            printf("Xmax = ");
        if(i==3)
            printf("Ymax = ");
        scanf("%d", &clip[i]);
    }

    printf("Enter the number of lines to be clipped: ");
    scanf("%d", &n);

    printf("Enter the coordinates of the line-endpoints:\n");
    for(i=0; i<n; i++)
    {
        printf("(%d)\n", i+1);
        for(j=0; j<4; j++)
            scanf("%d", &ln[i][j]);
    }

    initwindow(640, 480, "cohen");

    rectangle (clip[0], clip[1], clip[2], clip[3]);
    for(i=0; i<n; i++)
        line (ln[i][0], ln[i][1], ln[i][2], ln[i][3]);
    getch();

    cleardevice();

    rectangle (clip[0], clip[1], clip[2], clip[3]);
    for (i=0; i<n; i++)
    {
        cohen_sutherland (ln[i][0], ln[i][1], ln[i][2], ln[i][3], clip[0], clip[1], clip[2], clip[3]);
        getch();
    }
    return 0;
}

```

