Harsh Kashiwal
2003085
C21



Page No.
Date

# Experiment 8

**Aim:** Implement Sutherland Hodgeman polygon clipping method in C.

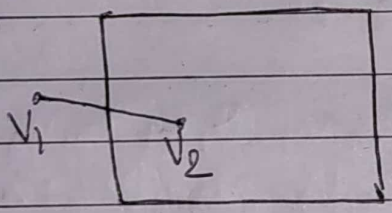
**Theory:**

**Algorithm:**

Step 1: Take input, vertices and edges of clip window. Also, create a list to store vertices.

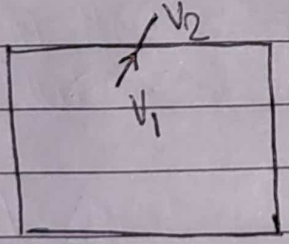
Step 2: Traverse through the edges of the polygon in order check which condition each edge falls into.

1) Outside → Inside.



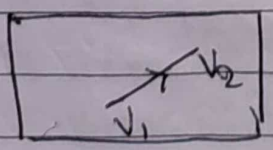

In this case, add intersection point & destination point to the list.

2) Inside → Outside.



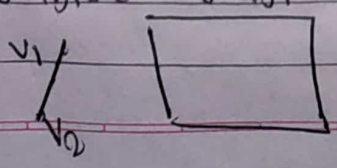

In this case, only add intersection point to the list.

3) Inside → Inside.




In this case, add destination point to the list.

4) Outside → Outside




Don't add

Page No.
Date

Step 3: Repeat step 2 for each edge of the clipping window, that is left, right, top and bottom, after clearing the list and considering the list as the new polygon.

Step 4: Traverse the vertices in the list in order and draw each edge.

## PROGRAM:

## Code:

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

float pts[20];
int count = 0;
int xmin, ymin, xmax, ymax;
void intersectX(float, float, float, float, float), intersectY(float, float, float, float, float);
void clipLeft(int, int, int, int), clipRight(int, int, int, int), clipTop(int, int, int, int), clipBottom(int, int, int, int);

void main()
{
	int gd = DETECT, gm;
	float tri[20]; //Original points of the triangle
	int i, n;
	initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
	printf("Enter the coordinates of your triangle as x1 y1 x2 y2 x3 y3\n");
	scanf("%f %f %f %f %f %f", &tri[0], &tri[1], &tri[2], &tri[3], &tri[4], &tri[5], &tri[6]);
	tri[6] = tri[0];
	tri[7] = tri[1];
	for(i = 0; i < 6; i+=2)
		line(tri[i], tri[i+1], tri[i+2], tri[i+3]);
```

```
	printf("Enter the coordinates of the clipping window as xmin ymin xmax ymax\n");
	scanf("%d %d %d %d", &xmin, &ymin, &xmax, &ymax);

	setcolor(RED);
	rectangle(xmin, ymin, xmax, ymax);

	setcolor(WHITE);

	printf("Press any button to show the clipped triangle...");
	getch();

	cleardevice();
	setcolor(RED);
	rectangle(xmin, ymin, xmax, ymax);

	count = 0;
	for(i = 0; i < 6; i+=2)
		clipLeft(tri[i], tri[i+1], tri[i+2], tri[i+3]);

	n = count;
	count = 0;
	for(i = 0; i < n; i++)
		tri[i] = pts[i];
	tri[i] = pts[0];
	tri[i+1] = pts[1];
```

```
for(i = 0; i < n; i+=2)
        clipBottom(tri[i], tri[i+1], tri[i+2], tri[i+3]);

n = count;
count = 0;
for(i = 0; i < n; i++)
        tri[i] = pts[i];
tri[i] = pts[0];
tri[i+1] = pts[1];

for(i = 0; i < n; i+=2)
        clipRight(tri[i], tri[i+1], tri[i+2], tri[i+3]);

n = count;
count = 0;
for(i = 0; i < n; i++)
        tri[i] = pts[i];
tri[i] = pts[0];
tri[i+1] = pts[1];

for(i = 0; i < n; i+=2)
        clipTop(tri[i], tri[i+1], tri[i+2], tri[i+3]);

n = count;
count = 0;
for(i = 0; i < n; i++)
        tri[i] = pts[i];
```

```
        tri[i] = pts[0];
        tri[i+1] = pts[1];

        setcolor(GREEN);
        for(i = 0; i < n; i+=2)
                line(tri[i], tri[i+1], tri[i+2], tri[i+3]);

        getch();
        getch();
}

void clipLeft(int x1, int y1, int x2, int y2)
{
        int i, initCount = count;
        if(x1 <= xmin && x2 >= xmin) //Outside to inside
        {
                intersectX(x1, y1, x2, y2, xmin);
                pts[count++] = x2;
                pts[count++] = y2;
        }
        else if(x1 >= xmin && x2 <= xmin) //Inside to outside
        {
                intersectX(x1, y1, x2, y2, xmin);
        }
        else if(x1 >= xmin && x2 >= xmin) //Inside to inside
        {
                pts[count++] = x2;
```

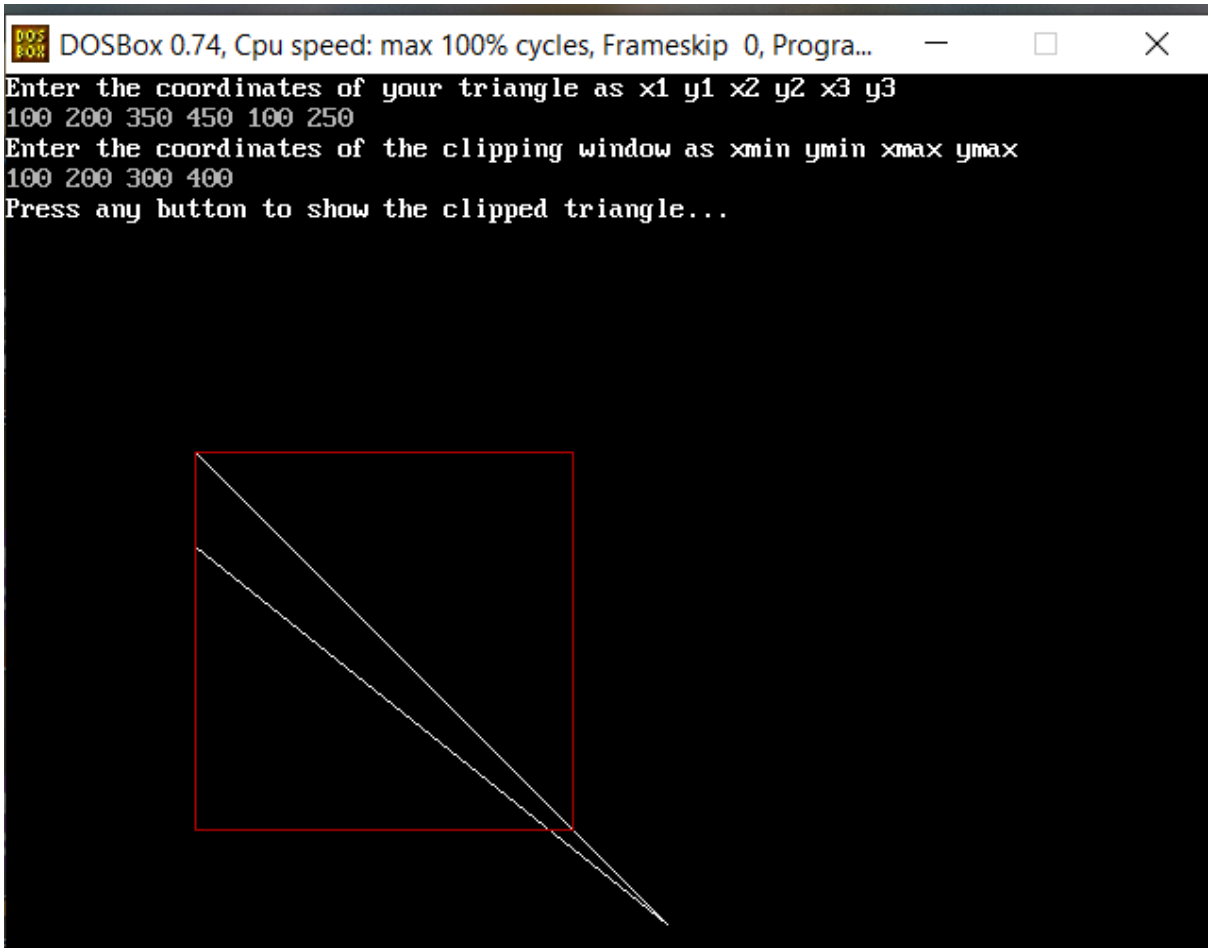
```
            pts[count++] = y2;
        }
        //else outside to outside, don't save anything
}

void clipRight(int x1, int y1, int x2, int y2)
{
        int i, initCount = count;
        if(x1 >= xmax && x2 <= xmax) //Outside to inside
        {
            intersectX(x1, y1, x2, y2, xmax);
            pts[count++] = x2;
            pts[count++] = y2;
        }
        else if(x1 <= xmax && x2 >= xmax) //Inside to outside
        {
            intersectX(x1, y1, x2, y2, xmax);
        }
        else if(x1 <= xmax && x2 <= xmax) //Inside to inside
        {
            pts[count++] = x2;
            pts[count++] = y2;
        }
        //else outside to outside, don't save anything
}

void clipTop(int x1, int y1, int x2, int y2)
```

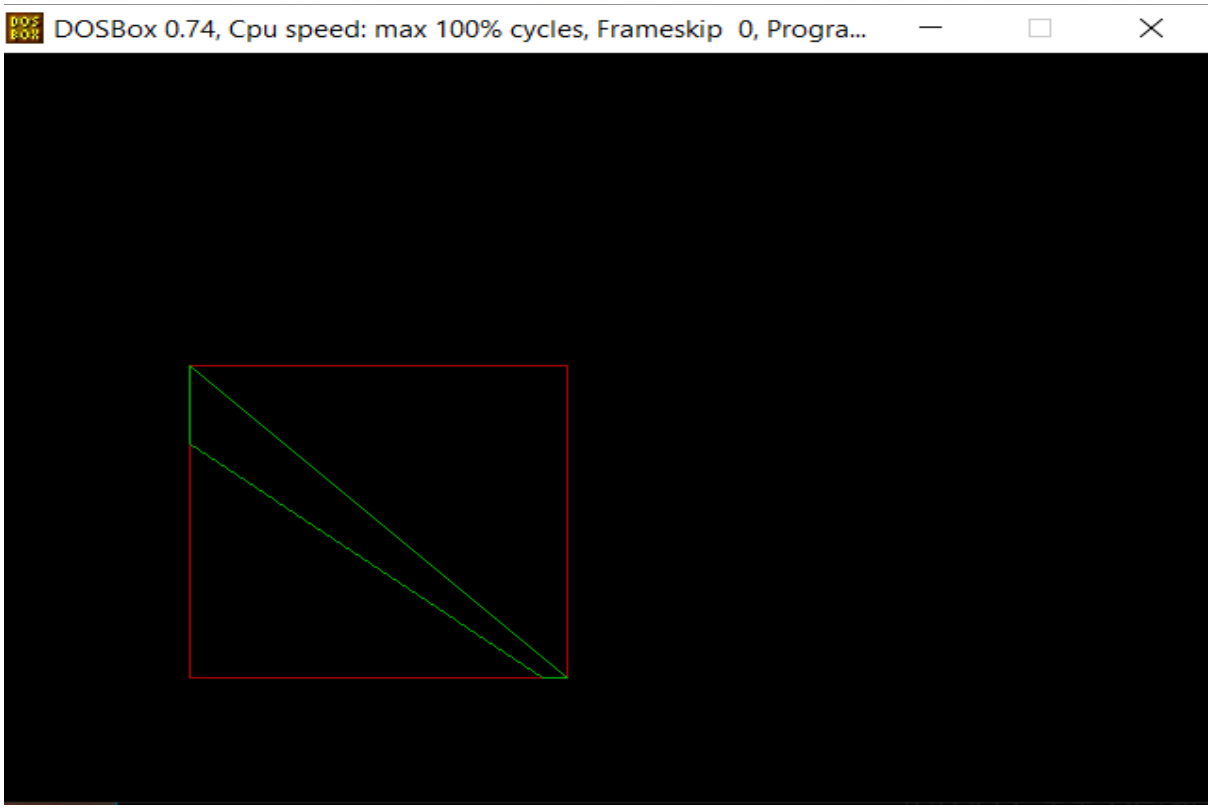
```
{
	int i, initCount = count;
	if(y1 <= ymin && y2 >= ymin) //Outside to inside
	{
		intersectY(x1, y1, x2, y2, ymin);
		pts[count++] = x2;
		pts[count++] = y2;
	}
	else if(y1 >= ymin && y2 <= ymin) //Inside to outside
	{
		intersectY(x1, y1, x2, y2, ymin);
	}
	else if(y1 >= ymin && y2 >= ymin) //Inside to inside
	{
		pts[count++] = x2;
		pts[count++] = y2;
	}
	//else outside to outside, don't save anything
}

void clipBottom(int x1, int y1, int x2, int y2)
{
	int i, initCount = count;
	if(y1 >= ymax && y2 <= ymax) //Outside to inside
	{
		intersectY(x1, y1, x2, y2, ymax);
		pts[count++] = x2;
```

```
            pts[count++] = y2;
        }
        else if(y1 <= ymax && y2 >= ymax) //Inside to outside
        {
            intersectY(x1, y1, x2, y2, ymax);
        }
        else if(y1 <= ymax && y2 <= ymax) //Inside to inside
        {
            pts[count++] = x2;
            pts[count++] = y2;
        }
        //else outside to outside, don't save anything
}

void intersectX(float x1, float y1, float x2, float y2, float x)
{
        //y = mx + c
        //=> y1 = mx1 + c
        //=> c = y1-mx1
        float m = (y2-y1)/(x2-x1), c = y1-m*x1;
        float y = m*x+c;
        pts[count++] = x;
        pts[count++] = y;
}

void intersectY(float x1, float y1, float x2, float y2, float y)
{
```

```
        float m = (y2-y1)/(x2-x1), c = y1-m*x1;
        float x = (y-c)/m;
        if(x2 == x1) //If slope is INF
        {
                x = x1;
        }
        pts[count++]=x;
        pts[count++]=y;
}
```

**OUTPUT:**


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...
Enter the coordinates of your triangle as x1 y1 x2 y2 x3 y3
100 200 350 450 100 250
Enter the coordinates of the clipping window as xmin ymin xmax ymax
100 200 300 400
Press any button to show the clipped triangle...


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Progra...