**POLYGON CLIPPING PROGRAM IN C | SUTHERLAND - HODGEMAN ALGORITHM FOR POLYGON CLIPPING**

In computer graphics, we have to study the Sutherland - Hodgeman polygon clipping algorithm. The algorithm according to the book by A. P. Godse is as follows:

**Algorithm:-**

       Step 1- Read coordinates of all the vertices of polygon

       Step 2- Read coordinates of the clipping window

       Step 3- Consider the left edge of window

       Step 4- Compare the vertices of each edge of the                                polygon , individually with clipping plane.

       Step 5-Save the resulting intersections and vertices in                         the new list of vertices according to four possible                     relationships between the edge and the clipping                     boundary discussed earlier.

       Step 6- Repeat the steps 4 and 5 for remaining edges of                     the clipping window. Each time the resultant list                       of the vertices of is successively passed to                               process the next edge of the clipping window.

       Step 7- Stop

Now for implementing this algorithm is so complicated and lengthy, so I've done some trick to get the same output but with different logic. You'll get the polygon clipped in the output.

Here's the program in c,

**Program:-**

|  |  |
| --- | --- |
|  | #include<stdio.h> |
|  | #include<graphics.h> |
|  | #include<conio.h> |
|  | #include<stdlib.h> |
|  | int main() |
|  | { |
|  | int gd,gm,n,\*x,i,k=0; |
|  | //window coordinates int wx1=220,wy1=140,wx2=420,wy2=140,wx3=420,wy3=340,wx4=220,wy4=340; |
|  | int w[]={220,140,420,140,420,340,220,340,220,140};//array for drawing window |
|  | detectgraph(&gd,&gm); |
|  | initgraph(&gd,&gm,"c:\\turboc3\\bgi"); //initializing graphics |
|  | printf("Window:-"); |
|  | setcolor(RED); //red colored window |
|  | drawpoly(5,w); //window drawn |
|  | printf("Enter the no. of vertices of polygon: "); |
|  | scanf("%d",&n); |
|  | x = malloc(n\*2+1); |
|  | printf("Enter the coordinates of points:\n"); |
|  | k=0; |
|  | for(i=0;i<n\*2;i+=2) //reading vertices of polygon |
|  | { |
|  | printf("(x%d,y%d): ",k,k); |
|  | scanf("%d,%d",&x[i],&x[i+1]); |
|  | k++; |
|  | } |
|  | x[n\*2]=x[0]; //assigning the coordinates of first vertex to last additional vertex for drawpoly method. |
|  | x[n\*2+1]=x[1]; |
|  | setcolor(WHITE); |
|  | drawpoly(n+1,x); |
|  | printf("\nPress a button to clip a polygon.."); |
|  | getch(); |
|  | setcolor(RED); |
|  | drawpoly(5,w); |
|  | setfillstyle(SOLID\_FILL,BLACK); |
|  | floodfill(2,2,RED); |
|  | gotoxy(1,1); //bringing cursor at starting position |
|  | printf("\nThis is the clipped polygon.."); |
|  | getch(); |
|  |  |
|  | cleardevice(); |
|  | closegraph(); |
|  | return 0; |
|  | } |

**Output:-**

After running this program, first you've to enter the number of vertices of polygon.

|  |
| --- |
| [polygon clipping output | enter number of vertices](https://1.bp.blogspot.com/-QTcTauMcqws/W7mF67aKE9I/AAAAAAAAFkI/wTjhfvl5EjcMVOYnPKWauFzDAKVa9z4pgCLcBGAs/s1600/1.PNG) |
| Output: Enter number of vertices |

After entering number of vertices you'll be asked to enter coordinates of vertices.

|  |
| --- |
| [Polygon clipping output | Enter coordinates of vertices](https://4.bp.blogspot.com/-4tXzG-FOdA0/W7mF-ApUHdI/AAAAAAAAFkQ/xDlN_RmjM_QR2uPUk1LzaciHbfP3XkggQCLcBGAs/s1600/2.PNG) |
| Output: Enter coordinates of vertices |

After entering vertices' coordinates just press a button a polygon will be drawn with white color. Now press a button to clip the polygon and you'll simply get the clipped polygon in the output.

|  |
| --- |
| [Polygon clipping output | Clipped polygon](https://4.bp.blogspot.com/-IOUvggXHD_w/W7mF91We6qI/AAAAAAAAFkM/1N6UShuqehsY1d57KXDWX2YKQvSUbP4pwCLcBGAs/s1600/3.PNG) |
| Clipped Polygon |