|  |  |
| --- | --- |
|  | **DEPARTMENT OF COMPUTER ENGINEERING** |



Semest

er



S.E. Semester III



–



Computer Engineering



Subj



Computer Graphics



Subject Professor In



-



char



Prof. Sachin Deshpande



Assisting

Teachers



Prof. Sachin Deshpande



|  |  |
| --- | --- |
| Student Name | Trisha Shah |
| Roll Number | 20102A0004 |

**EXPERIMENT DETAILS:**

|  |  |
| --- | --- |
| Title | Write a program to implement Sutherland Hodgeman Polygon Clipping |
| Tools/Language Used | Dev C++ |

**Sutherland Hodgeman Polygon Clipping**

It is performed by processing the boundary of polygon against each window corner or edge. First of all entire polygon is clipped against one edge, then resulting polygon is considered, then the polygon is considered against the second edge, so on for all four edges.

Four possible situations while processing

If the first vertex is an outside the window, the second vertex is inside the window. Then second vertex is added to the output list. The point of intersection of window boundary and polygon side (edge) is also added to the output line.

If both vertexes are inside window boundary. Then only second vertex is added to the output list.

If the first vertex is inside the window and second is an outside window. The edge which intersects with window is added to output list.

If both vertices are the outside window, then nothing is added to output list.

**Program:**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<stdlib.h>

#include<windows.h>

void gotoxy(short x, short y) //definition of gotoxy function//

{

COORD pos ={x,y};

SetConsoleCursorPosition(GetStdHandle(STD\_OUTPUT\_HANDLE), pos);

}

int main()

{

int gd,gm,n,\*x,i,k=0;

int w[]={220,140,420,140,420,340,220,340,220,140};

detectgraph(&gd,&gm);

initgraph(&gd,&gm,(char\*)"");

printf("Window:");

setcolor(RED);

drawpoly(5,w);

printf("Enter the no. of vertices of polygon:");

scanf("%d",&n);

x=(int \*)malloc(n\*2+1);

printf("Enter the coordinates of points:\n");

k=0;

for(i=0;i<n\*2;i+=2)

{

printf("(x%d,y%d): ",k,k);

int num=i+1;

scanf("%d %d",&x[i],&x[num]);

k++;

}

x[n\*2]=x[0];

x[n\*2+1]=x[1];

setcolor(WHITE);

drawpoly(n+1,x);

printf("\nPress a number to clip a polygon....");

scanf("%d",&k);

setcolor(RED);

drawpoly(5,w);

setfillstyle(SOLID\_FILL,BLACK);

floodfill(2,2,RED);

gotoxy(1,1);

printf("\nThis is the clipped polygon....");

getch();

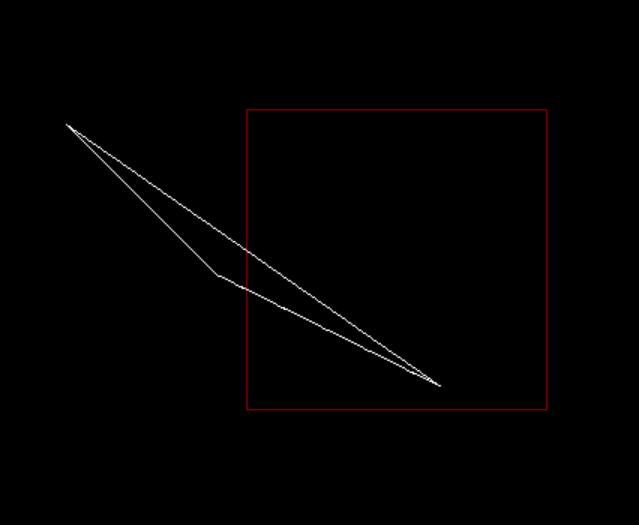
cleardevice();

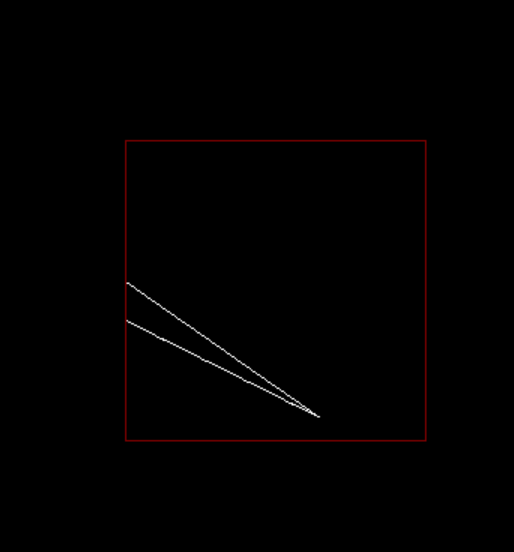
closegraph();

return 0;

}

**Output**

****

****

**Disadvantage of Cohen Hodgmen Algorithm:**

This method requires a considerable amount of memory. The first of all polygons are stored in original form. Then clipping against left edge done and output is stored. Then clipping against right edge done, then top edge. Finally, the bottom edge is clipped. Results of all these operations are stored in memory. So wastage of memory for storing intermediate polygons.