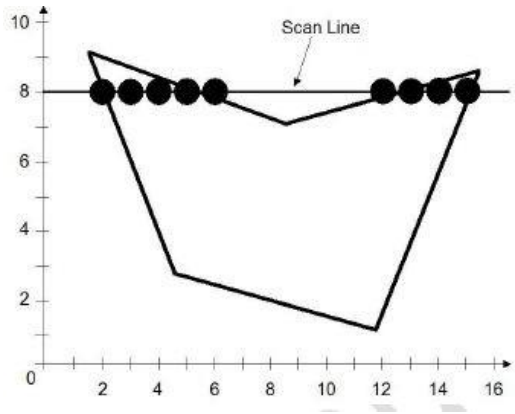
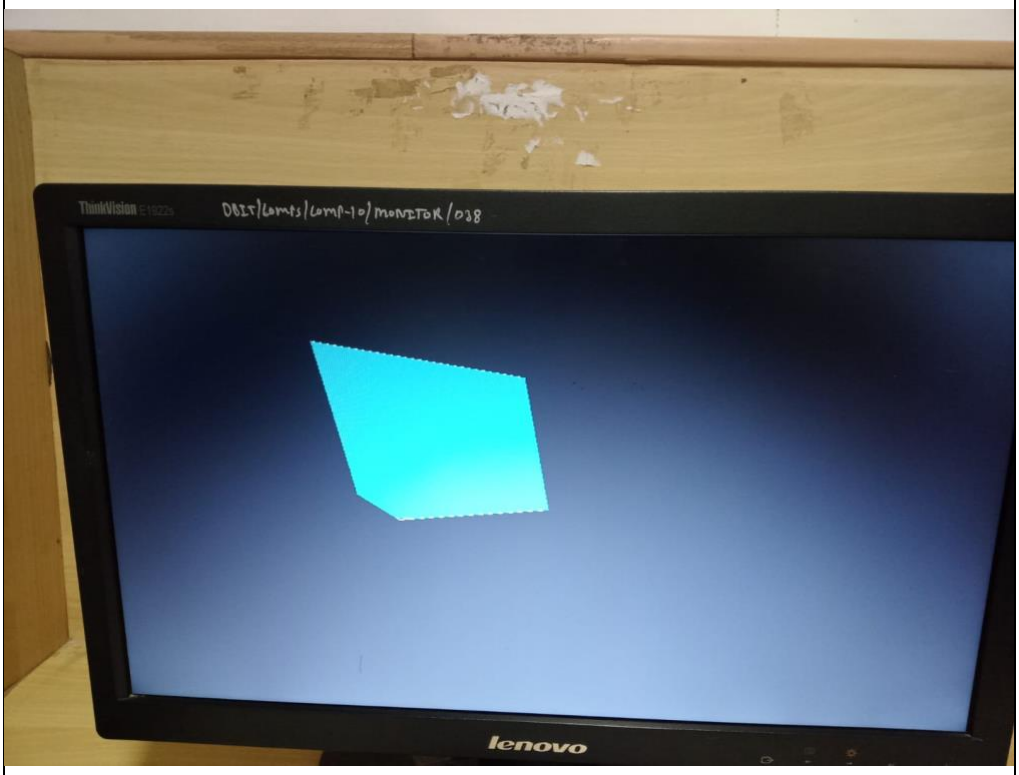
EXPERIMENT 7	
Title	**IMPLEMENTATION OF SCAN LINE FILL ALGORITHM**
Objective	To implement the Scan line polygon fill algorithm for coloring a given object.
Theory	**Description:** The basic scan-line algorithm is as follows: • Find the intersections of the scan line with all edges of the polygon • Sort the intersections by increasing x coordinate • Fill in all pixels between pairs of intersections that lie interior to the polygon **Process involved:** The scan-line polygon-filling algorithm involves • the horizontal scanning of the polygon from its lowermost to its topmost vertex, • identifying which edges intersect the scan-line, • and finally drawing the interior horizontal lines with the specified fill color process. 
Algorithm	1. the horizontal scanning of the polygon from its lowermost to its topmost vertex 2. identify the edge intersections of scan line with polygon 3. Build the edge table a. Each entry in the table for a particular scan line contains the

	maximum y value for that edge, the x-intercept value (at the lower vertex) for the edge, and the inverse slope of the edge. 4. Determine whether any edges need to be splitted or not. If there is need to split, split the edges. 5. Add new edges and build modified edge table. 6. Build Active edge table for each scan line and fill the polygon based on intersection of scanline with polygon edges.
Program	

```
#include <stdio.h>
#include <conio.h>
#include <graphics.h>
void main()
{
int n,i,j,k,gd,gm,dy,dx;
int x,y,temp;
int a[20][2],xi[20];
float slope[20];
clrscr();
printf("\n\n\tEnter the no. of edges of polygon : ");
scanf("%d",&n);
printf("\n\n\tEnter the cordinates of polygon :\n\n\n ");
for(i=0;i<n;i++)
{
printf("\tX%d Y%d : ",i,i);
scanf("%d %d",&a[i][0],&a[i][1]);
}
a[n][0]=a[0][0];
a[n][1]=a[0][1];
detectgraph(&gd,&gm);
initgraph(&gd,&gm,"C:\\TurboC3\\BGI");
/*- draw polygon -*/
for(i=0;i<n;i++)
{
line(a[i][0],a[i][1],a[i+1][0],a[i+1][1]);
}
getch();
for(i=0;i<n;i++)
{


dy=a[i+1][1]-a[i][1];
dx=a[i+1][0]-a[i][0];
if(dy==0) slope[i]=1.0;
if(dx==0) slope[i]=0.0;
if((dy!=0)&&(dx!=0)) /*- calculate inverse slope -*/
{
slope[i]=(float) dx/dy;
}
}
```

```
for(y=0;y< 480;y++)
{
k=0;
for(i=0;i<n;i++)
{
if( ((a[i][1]<=y)&&(a[i+1][1]>y))||
((a[i][1]>y)&&(a[i+1][1]<=y)))
{
xi[k]=(int)(a[i][0]+slope[i]*(y-a[i][1]));
k++;
}
}
for(j=0;j<k-1;j++) /*- Arrange x-intersections in order -*/
for(i=0;i<k-1;i++)
{
if(xi[i]>xi[i+1])
{
temp=xi[i];
xi[i]=xi[i+1];
xi[i+1]=temp;
}
}
setcolor(3);
for(i=0;i<k;i+=2)
{
line(xi[i],y,xi[i+1]+1,y);
getch();
}
}
}
```


Output	
Conclusion	Thus a C Program to implement Scan line polygon fill algorithm for coloring a given object was written and executed.