PRACTICAL 6:

AIM: Write a Program to fill polygon using Flood Fill & Boundary Fill Algorithm.

Source Code:

```
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
#include<dos.h>
#include<graphics.h>
void boundaryFill4(int x, int y, int fill_color,int boundary_color)
{
   if(getpixel(x, y) != boundary_color &&
     getpixel(x, y) != fill_color)
   {
         putpixel(x, y, fill_color);
         boundaryFill4(x + 1, y, fill_color, boundary_color);
         boundaryFill4(x, y + 1, fill_color, boundary_color);
         boundaryFill4(x - 1, y, fill_color, boundary_color);
         boundaryFill4(x, y - 1, fill_color, boundary_color);
   }
}
void flood(int x, int y, int new_col, int old_col)
{
   // check current pixel is old_color or not
   if (getpixel(x, y) == old_col) {

         // put new pixel with new color
         putpixel(x, y, new_col);

         // recursive call for bottom pixel fill
         flood(x + 1, y, new_col, old_col);

         // recursive call for top pixel fill
         flood(x - 1, y, new_col, old_col);

         // recursive call for right pixel fill
         flood(x, y + 1, new_col, old_col);

         // recursive call for left pixel fill
         flood(x, y - 1, new_col, old_col);
   }
}
```

```
void main()
{
        int gd,gm;
        int arr[14]={120,150,120,170,170,200,200,170,200,150,170,150,120,150};
        int arr1[]={50,50,50,100,100,100,100,50,50,50};
        int x=121,y=155,a=55,b=55;
        gd=DETECT;
        initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

        drawpoly(7,arr);
        delay(10);
    //  boundaryFill4(x, y, 6, 15);
        flood(x, y,4,0);

        drawpoly(5,arr1);
        boundaryFill4(a,b,3,WHITE);
        getch();
        closegraph();
}
```

Output: