

**PRACTICAL 9**

a. Write a program to fill a circle using Flood Fill Algorithm.

A1)

**4-connected****CIRCLE FLOOD FILLED BY 4-CONNECTED****CODE:**

```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
void flood(int,int,int,int);
void main()
{
    int gd=DETECT,gm;
    clrscr();
    initgraph(&gd,&gm,"C:\\\\TURBOC3\\\\BGI");
    printf("\nCircle filled by flood fill algorithm");
    circle(100,200,40);
    flood(104,204,8,0);
    getch();
}
void flood(int x,int y,int fill_col,int old_col)
{
    if((getpixel(x,y)==old_col))
    {
        delay(1);
        putpixel(x,y,fill_col);
        flood(x+1,y,fill_col,old_col);
        flood(x-1,y,fill_col,old_col);
        flood(x,y+1,fill_col,old_col);
        flood(x,y-1,fill_col,old_col);
    }
}
```

**OUTPUT:**



A2)

### 8-connected

#### RECTANGLE FLOOD FILLED BY 8-CONNECTED

##### CODE:

```
#include<conio.h>
#include<stdio.h>
#include<graphics.h>
void flood(int,int,int,int);
void main()
{
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"C:\\TC\\BGI");
    printf("Rectangle filled by flood fill: 8-conecter");
    rectangle(50,80,100,120);
    flood(54,104,8,0);
    getch();
}
void flood(int x,int y,int fill_col,int old_col)
{
    if((getpixel(x,y)==old_col))
    {
        delay(1);
        putpixel(x,y,fill_col);
        flood(x+1,y,fill_col,old_col);
        flood(x-1,y,fill_col,old_col);
        flood(x,y+1,fill_col,old_col);
```

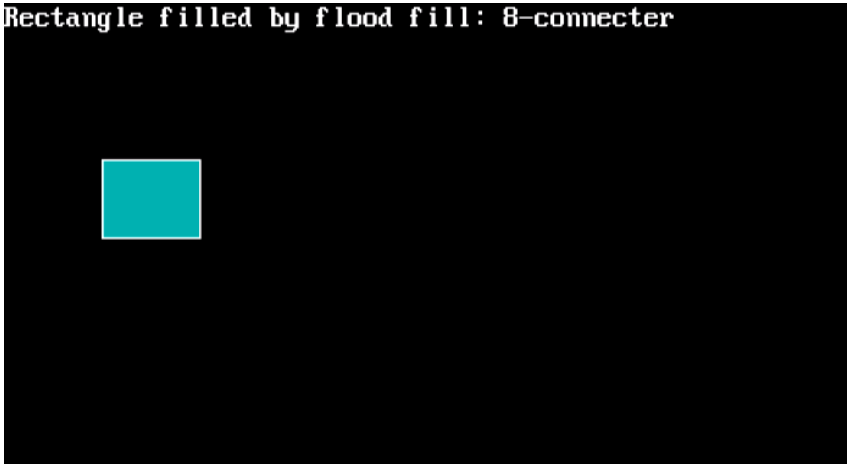
```

        flood(x,y-1,fill_col,old_col);
        flood(x+1,y-1,fill_col,old_col);
        flood(x+1,y+1,fill_col,old_col);
        flood(x-1,y-1,fill_col,old_col);
        flood(x-1,y+1,fill_col,old_col);
    }
}

```

**OUTPUT:**

Rectangle filled by flood fill: 8-connector



b. Write a program to fill a circle using Boundary Fill Algorithm.

B1)

**4-connected**

**CIRCLE FILLED BY BOUNDARY FILLED 4-CONNECTED**

**CODE:**

```

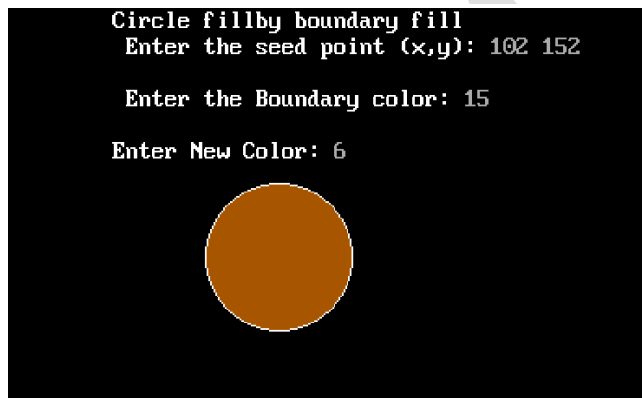
#include<conio.h>
#include<stdio.h>
#include<graphics.h>
void boundary_fill(int x,int y,int fcolor,int bcolor)
{
    if((getpixel(x,y)!=bcolor)&&(getpixel(x,y)!=fcolor))
    {
        putpixel(x,y,fcolor);
        boundary_fill(x+1,y,fcolor,bcolor);
        boundary_fill(x-1,y,fcolor,bcolor);
        boundary_fill(x,y+1,fcolor,bcolor);
        boundary_fill(x,y-1,fcolor,bcolor);
    }
}

```

```

    }
}
void main()
{
    int x,y,fcolor,bcolor;
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
    printf("Circle fillby boundary fill: ");
    printf("\n Enter the seed point (x,y): ");
    scanf("%d%d",&x,&y);
    printf("\n Enter the Boundary color: ");
    scanf("%d",&bcolor);
    printf("\nEnter New Color: ");
    scanf("%d",&fcolor);
    circle(102,152,45);
    boundary_fill(x,y,fcolor,bcolor);
    getch();
}

```

**OUTPUT:****B2)****8-connected****RECTANGLE FILLED BY BOUNDARY FILLED 8- CONNECTED**

```

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

```

**Komal Jambhale**

```

void boundary_fill(int x,int y,int fcolor,int bcolor)
{
    if((getpixel(x,y)!=bcolor)&&(getpixel(x,y)!=fcolor))
    {
        delay(1);
        putpixel(x,y,fcolor);
        boundary_fill(x+1,y,fcolor,bcolor);
        boundary_fill(x-1,y,fcolor,bcolor);
        boundary_fill(x,y+1,fcolor,bcolor);
        boundary_fill(x,y-1,fcolor,bcolor);
        boundary_fill(x+1,y+1,fcolor,bcolor);
        boundary_fill(x+1,y-1,fcolor,bcolor);
        boundary_fill(x-1,y+1,fcolor,bcolor);
        boundary_fill(x-1,y-1,fcolor,bcolor);
    }
}

void main()
{
    int x,y,fcolor,bcolor;
    int gd=DETECT,gm;
    initgraph(&gd,&gm,"C:\\TC\\BGI");
    outtextxy(50,50,"RECTANGLE FILLED BY FLOOD FILL:8-CONNECTER");
    rectangle(50,80,100,120);
    boundary_fill(52,82,15,6);
    getch();
}

```

