9. Solve the following

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

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

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

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<dos.h>
void flodfill(int x,int y,int f,int o)
{
int c;
 c=getpixel(x,y);
  if(c==o)
  {
  setcolor(f);
  putpixel (x,y,f);
   delay(10);
   flodfill(x+1,y,f,o);
   flodfill(x,y+1,f,o);
   flodfill(x+1,y+1,f,o);
   flodfill(x-1,y-1,f,o);
   flodfill(x-1,y,f,o);
   flodfill(x,y-1,f,o);
   flodfill(x-1,y+1,f,o);
   flodfill(x+1,y-1,f,o);
  }
}
void main()
{
int gd=DETECT,gm;
initgraph(&gd,&gm,"c:\\tc\\bgi");
rectangle(50,50,100,100);
flodfill(51,51,4,0);
getch();
}
```

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

Solution:-

```
#include<graphics.h>
```

```
}

#include<dos.h>
#include<conio.h>
void boundaryFill8(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);
        boundaryFill8(x + 1, y, fill_color, boundary_color);
        boundaryFill8(x, y + 1, fill_color, boundary_color);
        boundaryFill8(x - 1, y, fill_color, boundary_color);
        boundaryFill8(x, y - 1, fill_color, boundary_color);
        boundaryFill8(x - 1, y - 1, fill_color, boundary_color);
        boundaryFill8(x - 1, y + 1, fill_color, boundary_color);
        boundaryFill8(x + 1, y - 1, fill_color, boundary_color);
        boundaryFill8(x + 1, y + 1, fill_color, boundary_color);
    }

}
```

```
void main()

{

   int gd = DETECT, gm;

   initgraph(&gd, &gm, "c:\\Turboc3\\bgi");

   // Rectangle function
   rectangle(50, 50, 100, 100);
   // Function calling
   boundaryFill8(55, 55, 4, 15);
   delay(10000);
   getch();
/*closegraph function closes the graphics mode and
deallocates all memory allocated by graphics system .*/
   closegraph();
}
```