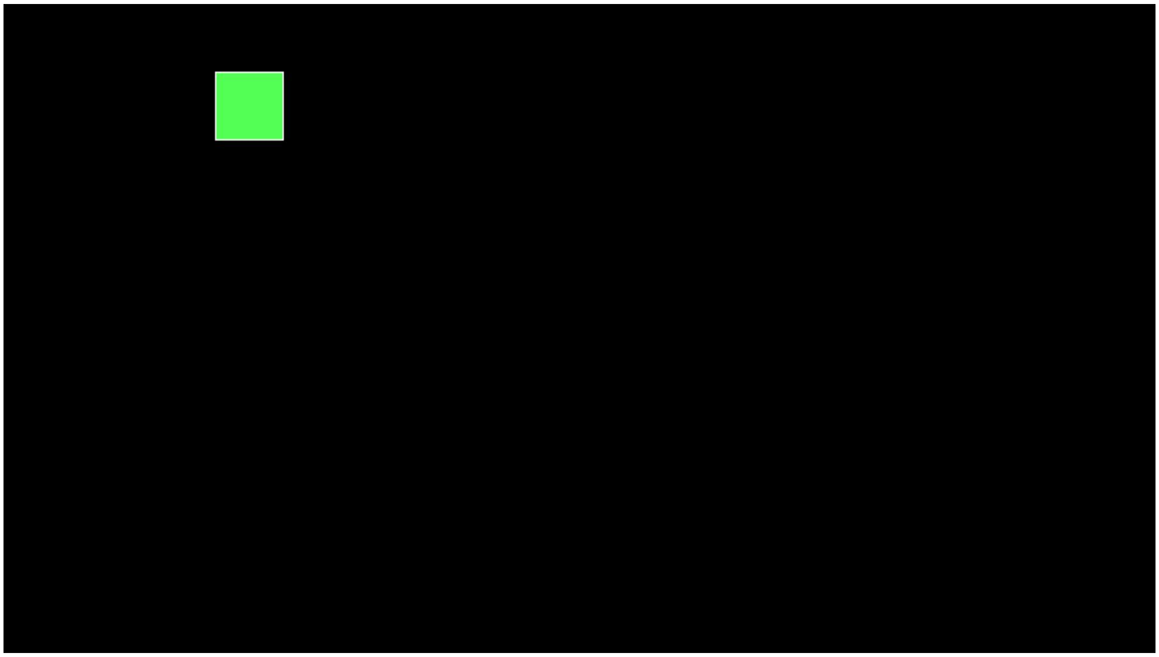
Implement Area Filling Algorithm : Boundary Fill, Flood Fill

1) Boundary Fill

```
#include<math.h>

#include<graphics.h>

#include<conio.h>

void boundary_fill(int x,int y,int f_col,int b_col)

{

if(getpixel(x,y)!=b_col && getpixel(x,y)!=f_col)

{

putpixel(x,y,f_col);

boundary_fill(x+1,y,f_col,b_col);

boundary_fill(x-1,y,f_col,b_col);

boundary_fill(x,y+1,f_col,b_col);

boundary_fill(x,y-1,f_col,b_col);

boundary_fill(x-1,y-1,f_col,b_col);

boundary_fill(x+1,y-1,f_col,b_col);

boundary_fill(x-1,y+1,f_col,b_col);

boundary_fill(x+1,y+1,f_col,b_col);

}

}

int main()

{
```

```
int gm, gd=DETECT;

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

rectangle(50,50,100,100);

boundary_fill(60,61,10,15);

getch();

closegraph();

return 0;

}
```



2) Flood Fill

```
#include<stdio.h>

#include<graphics.h>

#include<dos.h>
```

```
void flood(int,int,int,int);

int main()

{

int gd,gm=DETECT;

//detectgraph(&gd,&gm);

initgraph(&gd,&gm," ");

rectangle(50,50,100,100);

flood(55,55,12,0);

closegraph();

return 0;

}

void flood(int x,int y, int fill_col, int old_col)

{

if(getpixel(x,y)==old_col)

{

delay(10);

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);

}

}
```