LAB SET 10 (a)

Write a program to fill the a square like structure with Flood Fill algorithm

```
//Lab Set 10 (a)
//Akarsh Singh

//Flood filling a square
#include<stdio.h>
#include <GL/glut.h>

float fgcolor[] = { 0.0,0.0,1.0 };
float bgcolor[] = { 1.0,1.0,1.0 };
float colr[3];

void init()
{
    glClearColor(1, 1,1, 0.0);
    glColor3fv(fgcolor);
    glPointSize(2.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 640, 0, 480);
}

float* getPixelColor(float x, float y)
{

    glReadPixels(x, y, 1, 1, GL_RGB, GL_FLOAT, colr);
    return colr;
}

void setPixelColor(float x, float y)
{
    glColor3fv(fgcolor);
    glBegin(GL_POINTS);
    glVertex2f(x, y);
    glEnd();
    glFlush();
}

void floodFill(float x,float y)
{
    float *colr;
    colr = getPixelColor(x, y);
    if(*(colr+0)==bgcolor[0]&&*(colr+1)==bgcolor[1]&&*(colr+2)==bgcolor[2])

    {
        //printf("current pixel\n");
        setPixelColor(x, y);
        floodFill(x+1, y);
```

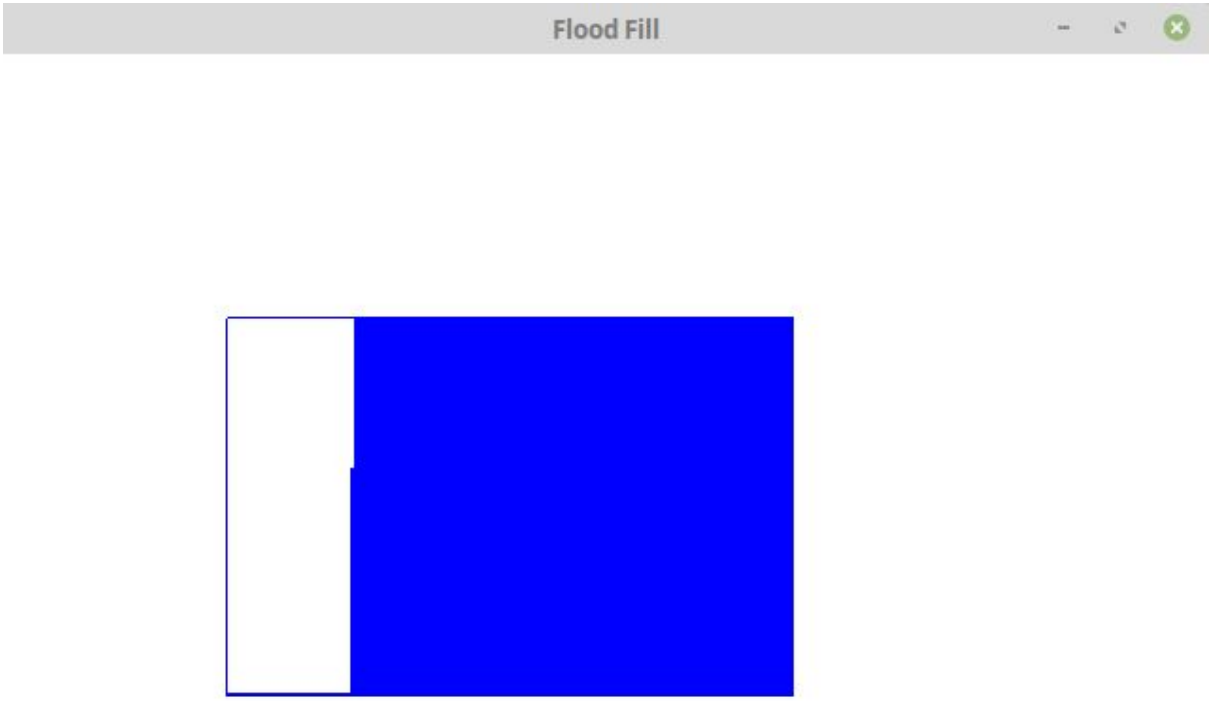
```
        floodFill(x-1, y);
        floodFill(x, y-1);
        floodFill(x, y+1);
    }
}

void onMouseClick(int button, int state, int x, int y)
{
    if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
        floodFill(121.0,141.0);
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINE_LOOP);
    glVertex2f(120.0,340.0);
    glVertex2f(420.0,340.0);
    glVertex2f(420.0,140.0);
    glVertex2f(120.0,140.0);
    glEnd();
    glFlush();
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE);
    glutInitWindowSize(640, 480);
    glutCreateWindow("Flood Fill");
    init();
    glutDisplayFunc(display);
    glutMouseFunc(onMouseClick);
    glutMainLoop();
    return 0;
}
```

OUTPUT



Flood Filling a Square

LAB SET 10 (b)

Write a program to fill the a square like structure with Scan-Line Fill algorithm

```
//Lab Set 10 (b)
//Akarsh Singh
//Scan-Line Fill
#include<stdio.h>
#include <GL/glut.h>

float colr[3];
float fgcolor[] = { 0.0,0.0,1.0 };
float bgcolor[] = { 1.0,1.0,1.0 };

void init()
{
    glClearColor(1, 1,1, 0.0);
    glColor3fv(fgcolor);
    glPointSize(2.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 640, 0, 480);
}

float* getPixelColor(float x, float y)
{
    glReadPixels(x, y, 1, 1, GL_RGB, GL_FLOAT, colr);
    return colr;
}

void setPixelColor(float x, float y)
{
    glColor3fv(fgcolor);
    glBegin(GL_POINTS);
    glVertex2f(x, y);
    glEnd();
    glFlush();
}

void scanFill()
{
    float *colr;
    float x,y;
    for(y=140.0;y<340;y++)
    {
        for(x=120.0;x<420;x++)
        {
            colr = getPixelColor(x, y);
```

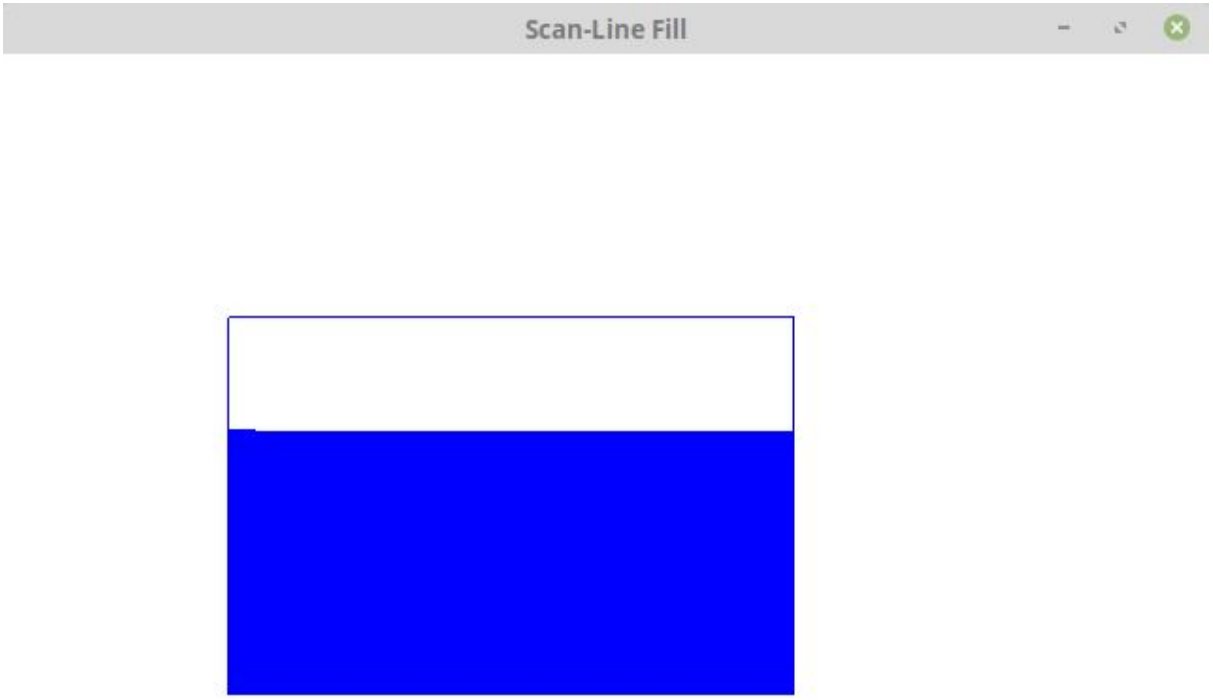
```
            if(*(colr+0)==bgcolor[0]&&*(colr+1)==bgcolor[1]&&*(colr+2)==
            bgcolor[2])
            {
                setPixelColor(x, y);
            }
        }
    }
}

void onMouseClick(int button, int state, int x, int y)
{
    if(button==GLUT_LEFT_BUTTON && state==GLUT_DOWN)
      scanFill();
}

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glBegin(GL_LINE_LOOP);
    glVertex2f(120.0,340.0);
    glVertex2f(420.0,340.0);
    glVertex2f(420.0,140.0);
    glVertex2f(120.0,140.0);
    glEnd();
    glFlush();
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE);
    glutInitWindowSize(640, 480);
    glutCreateWindow("Scan-Line Fill");
    init();
    glutDisplayFunc(display);
    glutMouseFunc(onMouseClick);
    glutMainLoop();
    return 0;
}
```

OUTPUT



Scan-Line Filling a Square