```
#include <iostream>
#include <math.h>
#include <time.h>
#include <GL/glut.h>
using namespace std;
void delay(float ms) {
clock_t goal = ms + clock();
while (goal > clock());
}
void init() {
glClearColor(0.0, 0.0, 0.0, 0.0);
glMatrixMode(GL_PROJECTION);
gluOrtho2D(0, 640, 0, 480);
}
void flood_it(int x, int y, float *fillColor, float *ic) {
float color[3];
glReadPixels(x, y, 1.0, 1.0, GL_RGB, GL_FLOAT, color);
if (color[0] == ic[0] && color[1] == ic[1] && color[2] == ic[2]) {
glColor3f(fillColor[0], fillColor[1], fillColor[2]);
glBegin(GL_POINTS);
glVertex2i(x, y);
glEnd();
glFlush();
flood_it(x - 2, y, fillColor, ic);
flood_it(x + 1, y, fillColor, ic);
flood_it(x, y + 1, fillColor, ic);
flood_it(x, y - 2, fillColor, ic);
}
}
void mouse(int btn, int state, int x, int y) {
y = 480 - y;
```

```
if (btn == GLUT_LEFT_BUTTON) {
if (state == GLUT_DOWN) {
float intCol[] = {1, 0, 0};
float color[] = {0, 0, 1};
glReadPixels(x, y, 1.0, 1.0, GL_RGB, GL_FLOAT, intCol);
flood_it(x, y, color, intCol);
}
}
}
void world() {
glPointSize(2);
glClear(GL_COLOR_BUFFER_BIT);
glColor3f(1, 0, 0);
glBegin(GL_POLYGON);
glVertex2i(15, 10);
glVertex2i(155, 200);
glVertex2i(305, 10);
glEnd();
glColor3f(0, 1, 0);
glBegin(GL_POLYGON);
glVertex2i(300, 398);
glVertex2i(150, 198);
glVertex2i(450, 198);
glEnd();
glColor3f(0, 0, 1);
glBegin(GL_POLYGON);
glVertex2i(300, 10);
glVertex2i(600, 10);
glVertex2i(450, 200);
glEnd();
glFlush();
```

```
}
int main(int argc, char **argv) {
glutInit(&argc, argv);
glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
glutInitWindowSize(640, 480);
glutInitWindowPosition(200, 200);
glutCreateWindow("Flood Fill");
glutDisplayFunc(world);
glutMouseFunc(mouse);
init();
glutMainLoop();
return 0;
}
```