```
#include<GL/gl.h>
#include<GL/glu.h>
#include<GL/glut.h>
#include<stdio.h>

typedef struct pixel{ GLubyte red, green, blue; } pixel;



void boundaryfill(float x,float y, pixel fill, pixel boundary)
{
  pixel c;
  glReadPixels(x, y, 1, 1, GL_RGB, GL_UNSIGNED_BYTE, &c);


 // printf("%d,%d,%d",(int)c.red,(int)c.green,(int)c.blue);
  if
((c.red!=boundary.red)&&(c.red!=boundary.blue)&&(c.green!=boundary.green)&&(c.green!
=fill.green)&&(c.blue!=fill.blue)&&(c.red!=fill.red)&&\
(x<=400))//&&(y<=100)&&(y>=50)&&(x>=200))
    {

      glBegin(GL_POINTS);
      glColor3ub(fill.red,fill.green,fill.blue);
      glVertex2f(x,y);
      glEnd();
      glFlush();
      glReadPixels(x, y, 1, 1, GL_RGB, GL_UNSIGNED_BYTE, &c);
      //printf("\nCOlOR %d,%d,%d",(int)c.red,(int)c.green,(int)c.blue);
 //printf("\nX=%f,Y=%f",x,y)


      boundaryfill(x+1,y,fill,boundary);
      boundaryfill(x-1,y,fill,boundary);
      boundaryfill(x,y+1,fill,boundary);
      boundaryfill(x,y-1,fill,boundary);

    }

}

void mydisplay()
{
      glBegin(GL_POLYGON);
      glColor3ub(10,10,10);
      glVertex2f(200,50);
      glVertex2f(200,100);
      glVertex2f(400,100);
      glVertex2f(400,50);
      glEnd();
      glFlush();
      pixel fill,boundary;
      fill.red=0;
      fill.green=0;
      fill.blue=255;
      boundary.red=255;
      boundary.green=255;
      boundary.blue=255;
      boundaryfill(300,75,fill,boundary);
      glEnd();
      glFlush();
}

void main(int argc,char **argv)
{
  glutInit(&argc,argv);
  glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
  glutInitWindowSize(400,400);
  glutInitWindowPosition(540,320);
  glutCreateWindow("my first attempt");
  glClearColor(1.0f,1.0f,1.0,0.0f);
  glClear(GL_COLOR_BUFFER_BIT);
  glutDisplayFunc(mydisplay);

  gluOrtho2D(0.0,400.0,0.0,400.0);
  glutMainLoop();


}
```