Confusing Code Makeover

This is a **pair** exercise and must be competed in your **tutorial** or **lab** with your partner.

During the **tutorial** this week, you will conduct a **code review** on the code below. You will have to figure out how the code works and what it does. Then you can discuss how it can be improved to make it more readable and to make it clearer.

During the **lab**, you will need download a copy of the file and fix it so that it works as expected and complies with the course **style guide**.

In the Lab

Step 1

With your lab partner, copy the confusing code shown in the section below into a file called **confusing.c**.

Try and figure out what this program is meant to do. Come up with some *test cases* you can use to make sure the code works as expected. Once you have some test cases, see if the program works properly.

Once you have some tests, you'll need to fix up the code so that is complies with the course **style guide** and so it passes your tests. How beautiful can you make the code?

When you fix the header comment, make sure to put your and your partner's name in, but don't remove Julian's.

As you fix up the code, keep some notes about what you have had to fix, why it needed fixing, and how you fixed it.

Step 2

At the end, make a final blog post with your version of the code as well as an explanation of what you had to change in order to make it comply with the **style guide** and what you had to change to make it clearer.

Step 3

Then, find the code of two other students in the course blog stream and comment on them stating:

- what you liked about their code makeover, and
- suggestions for how they could improve it for next time.

```
/*
* Test if a point is inside a triangle.
* Julian Saknussemm
*
* Given Three points of a triangle, and another arbitrary point this program
* determines if that point lies inside
* the triangle.
*
* This is determined by satisfying the following rule:
* A point P(x,y) is inside triangle A(x0,y0), B(x1,y1), C(x2,y2)
* iff
* P is on the same side of the line AB as C
* P is on the same side of the line BC as A
* and
* P is on the same side of the line AC as B
*
* A special case exits for a vertical line (inf gradient) when testing the side
*  of the line
*/

#include <stdio.h>

int test2( double px, double py, double m, double b )
{   if( py < m * px + b ) {
        return -1; // point is under line
           }else if ( py == m * px + b ){
           return 0;
           } else {
           return 1; // over
        }
}

int // two points lie on the same side of a line
test1( double px, double py, double m,double b, double lx,double
ly) {
        return (test2( px,py, m,b )==test2(lx,ly,m,b));
}
int tritest(double x0,double y0,double x1,double y1,double
x2,double y2,double px, double py){

        int line1, line2, line3;
// line eqns

double m01 = (y1-y0)/(x1-x0);
// b: y - y1 = m( x - x1 ), x = 0
double b01 = m01 * -x1 + y1;
double m02, m12, b02, b12;
m02 = (y2-y0)/(x2-x0);
m12 = (y2-y1)/(x2-x1);
```

```
b02 = m02 * -x2 + y2;
b12 = m12 * -x2 + y2;

// vertical line checks
if( x1 == x0 ) {
   line1 = ( (px <= x0) == (x2 <= x0) );
} else {
   line1 = test1( px, py, m01, b01,x2,y2);
}

if( x1 == x2 ) {
        line2 = ( (px <= x2) == (x0 <= x2) );
} else {
        line2 = test1(px,py, m12, b12,x0,y0);
}

if( x2 == x0 ) {
line3 = ( (px <= x0 ) == (x1 <= x0) );} else {
        line3 = test1(px, py, m02,b02,x1,y1);
}

   return line1 && line2 && line3;

}
int main(int argc, char* argv[]) {
double x0,y0,x1,y1,x2,y2,px;
double py;

int scanfsReturnValueAggregatedOverAllScanfs = 0;

// get input
        printf("Triangle Vertex A (x,y): ");
scanfsReturnValueAggregatedOverAllScanfs += scanf("%lf,%lf",
&x0,&y0);
        printf("Triangle Vertex  B (x,y): ");
scanfsReturnValueAggregatedOverAllScanfs += scanf("%lf,%lf",
&x1,&y1);
        printf("Triangle Vertex  C (x,y): ");
scanfsReturnValueAggregatedOverAllScanfs += scanf("%lf,%lf",
&x2,&y2);
   printf("Test Point (x,y): ");
scanfsReturnValueAggregatedOverAllScanfs += scanf("%lf,%lf",
&px,&py);

// print error
   if( scanfsReturnValueAggregatedOverAllScanfs != 8 ) {
                printf("You're stupid and didn't put in the right inputs!\n");
                return 1;
}
```

```
// print answer
   printf("Point (%.2lf,%.2lf) is ", px,py);
   if(tritest(x0,y0,x1,y1,x2,y2,px,py)) printf("inside");
   else printf("outside");
   printf(" the Triangel\n");

// return 0
return 0;
}
```

To run Styl-o-matic:

```
$ 1511 stylomatic confusing.c
Looks good!
```

You'll get advice if you need to make changes to your code.

Submit your work with the *give* command, like so:

```
$ give cs1511 wk03_confusingCode
```

Or, if you are working from home, upload the relevant file(s) to the `wk03_confusingCode` activity on Give Online.