CS 1713, Introduction to Computer Science
Assignment 3, Spring 1998
Sides and Angles of a Triangle
Due February 5, 1998

The assignment: The program for this assignment is similar to the copy programs of Assignments 1 and 2. This program should read *six* real numbers, either from the command line or from the standard input, in the same way that the copy program read three real numbers. The six numbers are to be interpreted as coordinates of three vertices of a triangle. Your program should compute the lengths of the three sides of this triangle, and in case the triangle is not degenerate (one or more sides of length zero), it should calculate and print the three angles, as well as calculating and printing the sum (to see that it is very close to 180 degrees).

Details:

1. Use the declarations

```
#define PI 3.14159265358979
double x1, y1, x2, y2, x3, y3;  /* coordinates of vertices  */
double a, b, c;  /* lengths of sides */
double A, B, C;  /* the three angles, A opposite a, etc. */
```

   for this assignment. Everything before the similar declarations in the copy program should stay the same, except of course that the initial comments should reflect this new assignment.

2. In order to read the six coordinates of three points of a triangle, you need to change the `a2, a1, a0` of the copy program to the list `x1, y1, x2, y2, x3, y3`. Just mimic the copy program, using the 6 names in the order shown instead of 3 names. Don't forget to change "`argc == 4`" to "`argc == 7`". Here, of course, `(x1, y1)` is the first point, etc.

3. To calculate the lengths of the sides, you need the formula for the distance between two points. For example, the distance between (x1, y1) and (x2, y2) is

   $$a = \sqrt{((x2 - x1)^2 + (y2 - y1)^2)},$$

   with similar formulas for the other two sides, `b` and `c`. Note that C does not directly have a "squaring" function, so you will need to write `(x2-x1)*(x2-x1)` for $(x2-x1)^2$.

4. If any of these sides have length zero, then the next formula will blow up due to division by zero. So just like the check for a2 equal zero in the copy program, we need to check for zero length sides here. In C, to check if "any of the sides have length zero", we can use

```
if (a == 0 || b == 0 || c == 0) {
    /* do something, say, print an error message */
    exit(1);
}
```

   This illustrates that "`==`" in C is "equals" and "`||`" in C is "or".

5. Here is a formula for the angle `A` opposite side `a`:

   $$A = \cos^{-1}((b^2 + c^2 - a^2)/(2\ b\ c)).$$

   You will have to translate this formula into the C language. $\cos^{-1}$ is the *arc cosine*, given in C by the function `acos`. Again you must write `b*b` for b^2, etc. There are similar formulas for angles `B` and `C`, opposite sides `b` and `c`, respectively.

6. The angles in step 5 are given in *radians*. Multiply by (180/π) to get degrees. Here you can use the number `PI` that was set to approximately the real π in step 1. Finally you need to add the three angles `A`, `B`, and `C`, getting the result in degrees. This sum should be close to 180.
7. As with the copy program, you should be able to enter the six numbers in either of two ways: either on the command line or after a prompt. Here is a sample interactive session of a solution to this assignment, showing the two methods of entering the six numbers. Your solution should look similar to this, but you should try several more commands. Notice below that the boldface is what you type in, the "`runner%`" prompts are produced by the system, and everything else is produced by the C program itself.

```
runner% triangle
This program was written by <your name here>
Sides and Angles of a Triangle
Enter x1, y1, x2, y2, x3, y3 (in order): 0.0 0.0 1.0 0.0 0.0 1.0
Vertices: (0.000,0.000), (1.000,0.000), (0.000,1.000)
Side lengths:  1.000, 1.000, 1.414
Angles:  45.00,  45.00,  90.00 degrees
Sum of angles: 180.00 degrees
runner% triangle 0.0  0.0  1.0  0.0  -1.0  1.0
This program was written by <your name here>
Vertices: (0.000,0.000), (1.000,0.000), (-1.000,1.000)
Side lengths:  1.000, 1.414, 2.236
Angles:  18.43,  26.57, 135.00 degrees
Sum of angles: 180.00 degrees
runner% triangle 0.0  0.0  1.0  0.0  1.0  0.0
This program was written by <your name here>
Vertices: (0.000,0.000), (1.000,0.000), (1.000,0.000)
Side lengths:  1.000, 1.000, 0.000
The triangle is degenerate
runner% triangle 0.0  0.0  1.0  0.0  -1.0  0.0
This program was written by <your name here>
Vertices: (0.000,0.000), (1.000,0.000), (-1.000,0.000)
Side lengths:  1.000, 1.000, 2.000
Angles:   0.00,   0.00, 180.00 degrees
Sum of angles: 180.00 degrees
```

8. You should name your program "`triangle.c`". Use a separate directory for your program, and use the following `makefile` in the directory, as before:

```
triangle:  triangle.c
        cc -g -o triangle triangle.c -lm
lint: triangle.c
        lint -m -u triangle.c -lm
```

9. Using `make` in step 8 and your file `triangle.c`, work with it until there are no more lint errors. (You might need help with some of these errors, and you might ignore some warnings while you try to compile and run.) Then compile using `make` as in step 8. (Here you may get some new error messages.) After all errors are gone, you can try runs until you get similar output to what is shown in step 7. (Here again you may need some help getting the right output. Remember to mimic the copy program and similar programs in your text.) Finally use the `script` command as in Assignment 1 to produce a file with the program `triangle.c` in it and with a transcript of a run similar to that in step 7 above. Use the final step of Assignment 1 to get a listing in the MS lab. Turn in this listing at the start of class on the due date, or at least by noon the next day, with your name on the top.