**PROBLEM**:

Write a function that finds both the inside radius and the outside radius of circles inside and outside a triangle. Your function must use **pointer notation** (the address operator “&” and the indirection operator “\*”). This function will be a file *separate* from the other code.

**The function prototype is**:

/\* Function to compute the semi\_perimeter of a triangle \*/

/\* and the radius of the inside circle and the outside circle \*/

**void find\_two\_radii (double a, double b, double c,**

**double\*radius\_inside, double\*radius\_outside);**

You will need the file **lab5.c** as your main/driver program for the function. This main program will set things up, read the values from the file, and print the output sentences.

You will also need **lab5.h** and **lab5.dat**.

**THE FORMULAS** (in ***algebraic*** notation)(must be translated to C notation):

(1) the semi-perimeter of a triangle.

s = 1/2 (a + b + c)

(2) the radius of a circle inscribed in a triangle of sides a, b, and c.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\/ (s – a)(s - b)(s - c)

radius = s

(3) the radius of a circle circumscribed about a triangle of sides a, b, and c.

abc

------------------------------

radius = 4 \/ s(s - a)(s - b)(s - c)

**TO GET THE FILES YOU NEED:**

First move to your class folder by typing: **cd csc60**

The following command will create a directory named **lab5** and put all the needed files into it below your csc60 directory.

Type: **cp -R /gaia/home/faculty/bielr/classfiles\_csc60/lab5 .**

Spaces needed: (1) After the **cp *↑*** *Don’t miss the space & dot.*

(2) After the **-R**

(3) After the directory name at the end & before the dot.

After the files are in your account and you are still in **csc60**, you need to type: **chmod 755 lab5**

This will give permissions to the directory.

Next move into lab5 directory, and type: **chmod 644 lab5\***

This will give permissions to the files.

Your new lab5 directory should now contain: lab5.c, lab5.h, lab5.dat

**INPUT/OUTPUT DESCRIPTION**:

The **input** is a list of varying length of type double values in the file **lab5.dat**. Each record (or line) will have three values for the three sides of the triangle. The data file will have four sets of values.

The **output** is a chart showing the three sides of the triangle and the two radii.

**DEFINED OUTPUT APPEARANCE**:

Print statements included in the main program, and require no changes, except for your name.

Your name here. Lab 5.

Triangle Sides Radius-Of-Circle

A B C Inside Outside

-------------------- ------ -------

3.70 5.00 4.20 0.46 2.55

ONLY the first line is shown here. You should validate the correctness of the other lines.

**ALGORITHM DEVELOPMENT - Pseudo code**:

/\*-------------------------------------------------------------------------\*/

main /\* main is given to you as ***lab5.c***  \*/

Open the data file and check for error on open.

Open the output file and check for error on open.

Print headers.

while ( good data reading(fscanf) the three sides)

Call find\_two\_radii

Print the sides and the two radii.

Close the files.

/\*-------------------------------------------------------------------------\*/

/\* This code will reside in a file **find\_two\_radii.c**, \*/

/\* *separate from the main function.* \*/

void find\_two\_radii(double a, double b, double c,

double \*radius\_inside, double\*radius\_outside)

Calculate the semi-perimeter

Calculate the inside radius

Calculate the outside radius

Return

/\*-------------------------------------------------------------------------\*/

**REMINDERS**:

* Remember to put your name and Lab 5 in the comment header and in the output.
* Avoid integer division. (1/2 can also be 1.0/2.0 or 0.5). Remember the operator for multiplication is the asterisk (\*).
* The use of *sqrt* requires the inclusion of *math.h* and, remember, that *sqrt* expects the argument (the number coming into it) to be a type double.
* You will also need to add **-lm** on the line that does the **gcc** in the makefile.
* You should examine the data file and confirm the correctness of the answer produced by your program.

**CREATING A MAKE FILE:**  Use the slides 13-14 of 5-UNIX as a reference. Also pasted at the end of this file.

radii

lab5.o find\_two\_radii.o

lab5.c lab5.h find\_two\_radii.c lab5.h

First line: use a name like *radii*, followed by the \*.o files and the \*.h file

Second line: one or two tabs followed by the \*.o files and the rename of the executable

Third and Fourth lines: the \*o file name, followed by a colon, followed by the \*.h filename

Above and below the fourth text line, include empty lines.

**PREPARE YOUR FILE FOR GRADING:**

When all is well and correct,

Type: **script StudentName\_lab5.txt** [Script will keep a log of your session.]

Type: **cat lab5.h** to display the file.

Type: **cat lab5.c** to display the code file.

Type: **cat find\_two\_radii.c** to display the code in your function

Type: **cat makefile** to display the contents of the makefile.

Type: **touch lab5.h** to force a recompilation

Type: **make** to compile the code

Type: **radii** to run the program to show the output of the program

(or whatever name you used for the executable)

Type: **cat lab5.out** to see the output of your program

Type: **exit** to leave the script session

**Turn in your completed session:**

Go to Canvas and turn in your script session (StudentName\_lab5.txt).

**Helpful slides:**

**Slide 13:**

*/\* Last pass at a makefile: \*/*

>**cat makefile**

power2: power2.o compute.o *header-file-if-needed*

gcc power2.o compute.o -o power2 –lm

power2.o: power2.h

compute.o: power2.h

**Slide 14:**

***/\* Helpful Comments \*/***

When you enter **vim**, while in Command Mode, type:  **:set list**

This will show the non-printable characters:

^I = tab

$ = end of line

To create a tab on athena, you may have to hit the tab key **twice** in a row.

You will know you have a tab when you see the ^I.