Programming Assignment #2

Zipped file: Due by 9 am on Thursday, October 9, 2014

This is a short assignment to practice creating simple functions using iteration. You will also gain more experience using floating-point numbers. The program will approximate the value of pi (π) - where π is defined as the ratio of a circle's circumference to its diameter - using two different methods.

The first method calculates the area of a quarter-circle using rectangles. As the number of rectangles increases, the width of each rectangle decreases and gives a more accurate value for pi. Diagrams are posted on the webpage and the algorithm will be discussed in the lab.

The second method for computing the approximate value of pi uses a formula derived by Gottfried Leibniz, also known as "the father of calculus." This method uses an infinite series of additions and subtractions to approximate pi:

$$\frac{\pi}{4} = 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} + \cdots$$

Notice that this approximates pi/4. As the number of terms in the series increases, the approximation will be closer to the value of pi/4. After the summation of the terms, the value must be multiplied by 4 to arrive at the final approximation of pi.

The results of these functions will be used to compare the circle algorithm and Leibniz's algorithm. The table will look something like this:

Approximations for pi

Iterations	Circle Method	Leibniz Method
1	3.46410162	4.00000000
10	3.15241143	3.04183962
100	3.14193686	3.13159290
1000	3.14160354	3.14059265
10000	3.14159300	3.14149265
100000	3.14159266	3.14158265
1000000	3.14159265	3.14159165

Details

You are given a file called **main.c**, which includes the main function. There are two functions prototyped in that file – one for the Circle method and the other for Leibniz's method. You must implement both of these functions exactly as prototyped in another file named **pi.c**.

A sample command line to build this assignment looks like this:

gcc –**Werror** –Wall –Wextra –ansi –pedantic main.c pi.c –o pi.exe

If your code is valid, this will compile and link both files and produce an executable file named **pi.exe**.

This assignment will require you to perform some simple mathematical calculations such as calculating square roots. The **sqrt** function in the Standard C library returns the square root of a number as a double. To use the math functions, you must include **math.h** in your pi.c file., Since you will not be submitting main.c, you must implement the two functions in pi.c. **REPEAT: You will not submit main.c,** so any changes you make to main.c will not be seen by me. Also: NEVER **PUT a main() FUNCTION IN YOUR FILE. EVER.** (Unless instructed to do so …)

Ensure that you follow the guidelines for documenting your code. This includes the file header comment, function header comments, and in-line commenting.

Both functions will implement some sort of looping. For the first function (circle_pi), you ***must*** use a **for** loop. For the second function (leibniz_pi), you ***must*** use a **while** loop.

The time to complete this assignment should be less than 2 hours. However, don't feel bad if you send more time on it, especially if you are taking the time to code it in the best way possible. Remember, your code says a lot about the type of software developer you are going to be. Professional developers can look at code and make judgements – for better or worse – about the person who wrote the code.

What to submit
You must submit the C file (**pi.c**) and the signed checklist (that must accompany every assignment) in a zip file. **Do not submit any other files than the ones specified above**.

If you have forgotten how to submit files, the details are posted on the course web page and the course syllabus. Failure to follow instructions will result in a poor score on the assignment (and possibly a zero).

Make sure your name and other info is on all documents

Special Note
The due date/time posted is the positively latest you are allowed to submit your code. Since the assignments can easily be completed well before the deadline, you should strive to submit it as early as possible. If you wait until the deadline and you encounter unforeseen circumstances (like being sick, or your computer breaking down, or something else), you may not have any time to submit the assignment on time. Moral: Don't wait until the last day to complete your homework.