Assignment number 2 for Computer Architecture

The assignment is to create a program that adds the number 1/2 to itself a large number of times and adds the number 1/3 to itself a large number of times separately first using type float and then type double. It is to then compare the values of adding the numbers to multiplying 1/2 time the number of times added to compute the “same sum” in a different way. The program will also multiply 1/3 times the number of times 1/3 was added to itself to compute the “same sum” in a different way. The program will then compare these two methods at arrive for the same value and output the difference. Hint, the value for the ½’s will be the same for the smaller numbers of times, the 1/3’s will never be the same. The output from your program is to be to a \*.txt file which you are to turn in along with your code.

The program must first add the ½’s and 1/3’s using type float and compare to the value obtain using multiplication instead of addition. It is to do this for the following values:

1.000

10,000

100,000

1,000,000

10,000,000

100,000,000

1,000,000,000

The program should do this using two nested loops. The outer loop determines the number of times to add and starts at a 1000 and increments by multiples of 10 up to a 1,000,000,000. The inner loop does the addition of the numbers 1/2 and 1/3 (Note, there are to be two sums, one sum is for the ½’s and the other for the 1/3’s). When the inner loop finishes, compute the products that “should” be the same. Hint, they will not always be the same. Why? After the inner loop completes, output the two sums and the two products, and the differences between the sums and the products that “should be the same” to check for rounding errors. Hint, the ½’s will be good for at least a while, but the 1/3’s will always be off since 1/3 does not go into a float (or double) exactly. Then continue the outer loop. Remember to clear your variables for the sums before the start of the inner loop. Send this output to a text file. The top line of the output file should have your name and the assignment number.

Next the program repeats the above except that it uses type double instead of type float for the variables. This will require a second set of nested loops and different variables.

Have your name, the assignment number, and a brief description of the program in comments at the top, have comments in the program, use proper blocking, and make certain to test your program to verify that it is working properly.

This program is due September 20.

You may do this assignment in C/C++ or Java (if you want to use another language, clear it with me first). Note that Python will not work for this assignment as it does not do single precision.

This assignment is worth 10 points.

Along with your code and output file, turn in an observations file, stating how much time this program took and what you learned from it. Make certain that you have your name and the assignment number at the top of your observations file and the top of your output file. Note, that I want the code and output files, not screenshots. As always, turn the result in to the class website DropBox for the problem.

The output should look something like below (feel free to do output your way, but make it neat and readable if you want full credit):

Name: John Smith

Assignment 1

FLOAT

For 1000 iterations

one half addition: 500.0

one half multiplication: 500.0

the difference is 0.0

one third addition: 333.3341

one third multiplication: 333.33334

the difference is 7.6293945E-4

and so forth for the rest of the values for float and then over again for double precision values.

Note that the sums for the ½’s should be correct for a while. The sums for the 1/3’s will always be off, only a little at first (see above example) but then get worse with time.

Note that this (and all the assignments in this class) is an individual assignment, you are not to work with someone on it. It is OK to ask for and to give some help for a problem within the assignment, but the work must be your own. This will be the case for all of the assignments in this class.