CS 7A - Spring 2015 - Finding roots using the Babylonian algorithm. Due 2/19/15

Background Theory

The Babylonian algorithm is used to compute roots like $\sqrt{2}$ and $\sqrt[3]{5}$. Pseudo code for the Babylonian algorithm to find square roots looks like this:

1. Enter a number, A, whose square root is to be computed.
2. Choose an initial guess, x_0: it doesn't need to be very good.
3. Iterate the formula,
$$x_{i+1} \leftarrow \frac{\frac{A}{x_i} + x_i}{2}$$
until the difference between x_{i+1} and x_i is small enough. Use a tolerance like $A/10^{10}$

The reason this works is because (1) if $x_i = \sqrt{A}$ then $x_{i+1} = x_i = \sqrt{A}$ also (2) if $x_i < \sqrt{A}$ then $A/x_i > \sqrt{A}$ so x_{i+1} is closer to $\sqrt{A}$ than x_i...and vice versa.

1. Use a function to evaluate x_{i+1} and loop the iteratives until the difference is less than some small tolerance, say, $A/10^{10}$.

   The function should be called by a main function some thing like this,

```
//<your name here>
//Working the babylonian algorithm

#include <iostream>
#include <complex>

using namespace std;

double babylonian(double A) {
    // write code here for the Babylonian algorithm
        // return square root of A
}

int main() {
        // set up variables, as needed
    cout << "\nEnter a number and we'll compute the square root: ";
    while(cin>>A) {
        cout << "\nThe square root of " << A << " is " << babylonian(A);
        cout << "\nEnter a number and we'll compute the square root: ";

    }
}
```

   and it should report the number of iterations that the Babylonian algorithm required, producing output like this:

```
This program will find the positive square root
of a positive number you supply.
```

```
Please enter a number: 5
The Babylonian algorithm required 5 iterations.
The square root of 5 is approximately 2.23607

Please enter a number: 7
The Babylonian algorithm required 5 iterations.
The square root of 7 is approximately 2.64575

Please enter a number: 100000
The Babylonian algorithm required 13 iterations.
The square root of 100000 is approximately 316.228
```

2. Modify the program to compute the nth root. For this, use the iterations:

$$x_{i+1} \leftarrow \frac{\frac{A}{x_i^{N-1}} + (N-1)x_i}{N}$$

submit the code for #2 using your initials in the usual format: say GH_babylonian.cpp