asgn2: A Little Slice of PI Doc

Program is divided up into 7 .c files, 6 of which contain relatively simple code for solving for something according to the provided mathematical equation.

e.c:
Attempts to calculate pi using Euler's taylor series.
 For loop looping until most recent value added to the ongoing value of e is less than epsilon
  Calculate k! by multiplying the previous k! used and multiplying it by the current k
  Add 1/k! to the ongoing e value

bbp.c:
Attempts to calculate pi using the Bailey-Borwein-Plouffe formula.
 For loop looping until most recent value added to the ongoing value of pi is less than epsilon
  //Dividing up the function in half for simplicity
  Calculate the right hand side by plugging in the current loop number into the provided equation
  Calculate the left hand side by taking the previous left hand side value and dividing by 16 instead of manually calculating the exponents value each time
  Multiply the two sides together and add the result to the value of pi

euler.c:
Attempts to calculate pi using Euler's solution.
 For loop looping until most recent value added to the ongoing value of e is less than epsilon
  Calculate the most recent value by calculating 1/Loop*Loop and add it to the value of pi
 Multiply the value of pi by 6 and take the square root to complete Euler's solution of pi

madhava.c:
Attempts to calculate pi using the Madhava series.
 For loop looping until most recent value added to the ongoing value of pi is less than epsilon
  Calculate the numerator by dividing the previous numerator by -3 to imitate the exponent increasing
  Calculate the denominator by multiplying 2 to the current loop number and adding 1
  Set the most recent value to the numerator divided by the denominator
  Add the most recent value to the value of pi

newton.c:
Attempts to solve for the square root of x using Newton's method.
 For loop looping until the most recent calculated fraction is less than epsilon
  //Mostly copying from Long's provided code in the asgn2 pdf
  Set the most recent value equal to the value of the current square value
  Calculate the new square value by multiplying .5 times the the most recent value plus the input value divided by the most recent value

viete.c:
Attempts to calculate pi using Viete's formula.
 For loop looping until the most recent value is less than epsilon
  Calculate the numerator by taking the square root of the previous numerator plus 2
  Divide the numertor by 2 and multiply it by the current value of pi

mathlib-test.c:
Acts as the main file that runs all of the other files to properly test and compare them to math.h.
Start:
 Use getopt to check which characters the user inputted when running the program
  Will likely have to store the variables in some sort of array to ensure that the functions are called
  in the correct order
 For each character that the user specified run the corresponding test

-h prints a help message
-a runs all tests
-e runs e approximation test
-b tuns Bailey-Borwein-Plouffe pi approximation test
-m runs madhava pi approximation test
-r runs euler sequence pi approximation test
-v runs viete pi approximation test
-n runs newton-raphson square root approximation tests
Has to check values 0 through 10 in increments of .1
-s prints extra statistics for each other test
For each test the corresponding value is calculated using the homemade function which is then compared to the valu
e resulting from math.h
Each test should print in the format of "(homemade_function_name)() = 0.000000000000000, (real_function_name)
() = 0.000000000000000, diff = 0.000000000000000