

## Assignment 2 design

### Description of Program

This program consists of 8 files that, together, contain various methods of calculating mathematical constants such as e and pi. This program will not implement the constants from math.h, but will compare the final values calculated with the math library. The various methods will calculate this while the difference between the previous and next term of the summation are greater than epsilon, which is a constant defined in the given mathlib.h file. Since the various methods outlined to calculate these constants are in the form of summations, I will use for or while loops to implement these calculations.

### Files included:

- mathlib-test.c
  - This is the testing harness for the functions that will approximate the constants e and pi.
- e.c
  - This file will contain two functions, e() and e\_terms(). The first of which will approximate the value of e with the taylor series  $e = \sum_{k=0}^{\infty} (1/k!)$ . The second will return the number of iterations
- madhava.c
  - This file will contain two functions, pi\_madhava() and pi\_madhava\_terms(). pi\_madhava() will compute pi using the summation listed  $pi/\sqrt{12} = \sum_{k=0}^{\infty} (-3)^{-k}/(2k+1)$ . pi\_madhava\_terms() will return the number of iterations used for the approximation
- euler.c
  - This file will contain two functions, pi\_euler() and pi\_euler\_terms(). pi\_euler() will compute pi using the summation listed . pi\_euler\_terms() will return the number of iterations used for the approximation
- bbp.c
  - This file will contain two functions, pi\_bbp() and pi\_bbp\_terms(). bbp\_euler() will compute pi using the summation listed. pi\_bbp\_terms() will return the number of iterations used for the approximation
- viete.c
  - This file will contain two functions, pi\_viete() and viete\_terms(). pi\_viete() will compute pi using the summation listed. viete\_terms() will return the number of iterations used for the approximation
- newton.c
  - This file will contain two functions, sqrt\_newton() and sqrt\_newton\_terms(). sqrt\_newton() will compute the square root using the summation listed . sqrt\_newton\_terms() will return the number of iterations used for the approximation
- mathlib.h

- This is a given header file with function declarations.
- Makefile
  - This will contain all the code for building the program using clang
  - Will include make clean for removing all compiler generated files
  - Will link all the files to mathlib executable
- README.md
  - This is an explanation of proper programs usage and command line arguments written in markdown syntax
- Design.pdf
  - This is the document being currently viewed, it contains pseudocode explanations and description of the program.
- Writeup.pdf
  - This will contain a writeup of graphs displaying the difference between values reported by function and math libraries
  - Analysis and explanations for discrepancies and findings

### **Pseudocode:**

### **Notes**

- mathlib-test.c: This file has one function, main(), which takes two inputs, int argc and char \*\*argv.
  - This file contains the test harness for the rest of the functions
  - This file runs the rest of the functions and compares them with mathlib values.
- e.c:
  - This file contains the functions e() and e\_terms().
  - This file computes e until the step is less than epsilon, which is given in the mathlib function
  - e\_terms returns the amount of times the function has iterated
  - This uses the Taylor series expansion of  $e^x$
- sqrt.c:
  -