

Design Document of Assignment 2  
Zihua Li  
CSE 13S  
Prof. Darrell Long

The purpose of this program is to compute the fundamental constants  $e$  and  $\pi$ .

Both  $e$  and  $\pi$  can be calculated using Taylor series.

For  $\pi$ , it could be calculated using Taylor series expansion of  $\arctan 1$ , because  $\pi$  is equal to 4 times of  $\arctan 1$ .

Same core method for calculating  $e$ . But we do not need to substitute something like  $\arctan 1$  as for  $\pi$ ,  $e$  is just  $\sum x^n / n!$  divided by  $n$ 's factorial.

The hard thing is not using anything from `math.h` and the `factorial()` function. We could use a for loop that iterates until the number of iterations reaches the limit, that reinstates a factorial function.

I have created functions for calculating `sqrt`, `power`, and `remainder` in this program, in order to calculate  $\pi$  in each method.

`e.c` Using Taylor series to compute the value of  $e$ . Already stated.

`madhava.c` Using Madhava series provided in the assignment instructions file, I have implemented the algorithm into the program. You need `sqrt` and `power` functions to calculate the SUM.

`euler.c` Using the formula derived from Euler's solution to the Basel problem to compute the value of  $\pi$ . You need `power` and `sqrt` to calculate the SUM.

`bbp.c` Using Bailey-Borwein-Plouffe formula to compute the value of  $\pi$ . You need `power` and `sqrt` to calculate the SUM.

`viete.c` Using Viete's formula to compute the value of  $\pi$ . You need `sqrt` to calculate the infinite product. (infinite restricted by num of iterations)

`newton.c` Using Newton-Raphson method to compute square root of the input. You can use the `absolute` function provided in `mathlib.h`, iterations stops when difference is no longer bigger than `EPSILON` (also in `mathlib.h`), you need to create a `remainder` function to complete the iteration.

Every sub-program globalizes its terms count, so when outputting data it could be used by the `mathlib-test.c` main program.

`mathlib-test.c` This is the main program. It outputs the data and takes in options to produce data from different methods, as well as the terms count number upon request by putting option in the command. It uses `getopt()` to

get the options user inputted. And a lot of if-else conditions to make sure it outputs the data of need. It uses ASCII to determine the option 'opt'.