David Shonack

Design Document Assignment 2

Program Objective:

This Program will calculate pi and e using Madhava sequence, taylor series, euler, Bailey-Borwein-Plouffe formula, and Viete formula. The user will be prompted if they want to choose a specific function to calculate the approximate value of pi or e. This program will also take the difference of the calculated pi value and math.h library pi value to determine how accurate each approximation is.

Files to be included:

e():

- This function will calculate e using the taylor series

e.terms():

- This function will return the amount of values used.

bbp():

- This function will return an estimated value of pi by using Bailey-Borwein-Plouffe formula

bbp_terms():

- This function will return the number of terms that have been iterated

euler():

- This function will return pi using euler's solution series

euler_terms():

- This function will return the amount of iterations euler() functions loop compiles

madHava():

- This function will return pi by using the madhava series.

madHava_terms():

- As you might of guessed by now this function will return the amount of time we go through the series madHave in the Madhava function newton()

- This function will take a parameter and find the square root of that parameter and return that value. Surprisingly we don't have a newton_terms function so we will return our square rooted value and the amount of times this function is called.

Matlib.h file:

- This file will have the user interface of our math library

Matlib-test.c:

- This file will test all of our function

README.md

- This file will give a detailed description of the program as well as how to use it. Along with any other information the user needs to know.

Pseudo Code:

```
Power function(a,b)
    Accepts two parameters, the base value, and exponential value
        A for loop that will run the same amount of iterations as our exponential value
            Set Base_value will times equal the Base_Value
        End loop
        Return the Base Value

Square_root Function(x)
    Define double z and y
    While the absolute value of the difference of y and z is greater then epsilon then
        z is equal to y
        y equals ½ times(z + x/z)
    Exit the loop
    Return the value of y to the main

e function(x,k)
    Declare FactorialVar variable
    While the total is greater then epsilon
        Set t equal to k
        For loop as t is greater then i then
        As long as t is greater then h then
        Set factorial *= t to its current value
        Where t is going down in value after each interaction
    Then continue to run the loop
    Exit the loop
    Call to initialize the terms we are using e.terms(Counter);
    Return the Total value




Function madhava()
    While total is greater then epsilon then continue
    The total variable will equal the  madhava formula
    Total to keep checking the while loop
    Exit loop
    Return value back to main

Function euler()

    for loop using n as iteration var
        Total is plus equal to euler's formula.
    Exit loop
    Total will equal our total result
    Square_root(total)
    Return the total or final amount


Function bbp()

    While our total amount is greater then epsilon
        Total is equal to equation Bailey-Borwein-Plouffe formula
        Increment k

Function Vietes()
    While the total is greater then epsilon then continue look
        Total is times equal to a/2
        Increment a
```

Main function Matlib()

This function will take an int as a parameter and a character pointer and use a while loop to transverse the user input, depending on what the user inputs it will return that function or return an error message if it is not one of the display options.