**BIL 105E – Introduction to Scientific and Engineering Computing (C)**

**Spring 2014-2015**

**Homework 2**

Assignment Date: 06.04.2015

Due Date: **20.04.2015 17:00**

You will implement a **Scientific Calculator** program. Your program should run as follows:

* When the program executes, all operators are first listed on the screen with their corresponding characters in parantheses:

-----------------------------------------------------------------------------------------------------------------

The list of operations in this simple calculator:

sine (S), cosine (N), tangent (T), exponent (E)

power (W), absolute (A), factorial (F)

add (+), subtract (-), divide(/), multiply(\*), modulus (%)

π (P)

Quit (Q)

Memory (M)

Clear (C)

Memory Clear (MC)

Memory Return (MR)

------------------------------------------------------------------------------------------------------------------

* Memory (M) option stores the result buffer in the memory.
* Clear (C) option clears the result buffer.
* Memory Return (MR) returns the stored result in the memory.
* Quit (Q) terminates the program.
* The user will enter the characters written in parantheses when they would like to perform a specific operator.
* Each operator should be implemented as a separate function in your code. You are required to implement these functions yourself, i.e., *without using any* *built-in functions from C libraries*. **(Mathematical functions that calculate sine, cosine, exponent etc. are not allowed. However you can use string functions from C libraries.)**
  + You need to define Taylor series to make the calculations of **sine, cosine** functions. All trigonometric functions will accept **Degrees** (not “Radians” or “Grads”). **Degrees can be decimal numbers as well as integers.**
  + When the user enters P, the program will print the value of *π*.
  + **You can assume that the exponential function takes a base 10 instead of *e.***
  + **In power function, you can assume that the power can take integers only.**
* All exceptional cases, such as *very large numbers entered by the user, decimal numbers, very large numbers calculated as the output, negative numbers,* etc., should be handled by your program.
  + Numbers with decimal points will be provided with a ‘.’ sign (example: 100.2).
  + The user input could be one of the following:
    - a letter such as P, Q, M, etc.
    - a negative number such as -123
    - a decimal number such as -25.456
    - a large number 1234456.7
  + Your program needs to understand if the user input is one of these shown above.
  + When an errorenous input is given by the user, the program will not terminate, but will prompt the user for another entry.
* Your program should be well commented and understandable.
* A sample execution can be seen below:

Enter user input: 30

Enter user input: S

Sin(30) = 0.5

Enter user input: M

Enter user input: MR

0.5

Enter user input: \*

Enter user input: 4

0.5 \* 4 = 2.0

Enter user input: E

Enter user input: 8

2 \* exp(8) = 200,000,000

Enter user input: M

Enter user input: MR

200,000,000

Enter user input: MC

Enter user input: MR

0

Enter user input: 8

Enter user input: F

Factorial(8) = 40320

Enter user input: 4

Enter user input: /

Enter user input: 2.5

4/2.5 = 1.6

Enter user input: \*

Enter user input: -45

1.6 \* -45 = -72.0

Enter user input: W

Enter user input: 2

-72^2= 5184

Enter user input: %

Enter user input: /

Invalid input for the operator (%).

Enter user input: 3

5184%3 = 0

Enter user input: Out

Invalid character. Choose from the operators below:

----------------------------------------------------------

sine (S), cosine (N), tangent (T), exponent (E)

power (W), absolute (A), factorial (F)

add (+), subtract (-), divide(/), multiply(\*), modulus (%)

π (P)

Quit (Q)

Memory (M)

Clear (C)

Memory Clear (MC)

Memory Return (MR)

----------------------------------------------------------

Enter user input: Q