Computing Machinery I
Assignment 6

File I/O and Floating-Point Numbers

Write an ARMv8 assembly language program to compute the functions e^x and e^{-x} using the series expansions shown below. Use double precision floating-point numbers. The program will read a series of input values from a file whose name is specified at the command line. The input values will be in binary format; each number will be double precision (and thus each is 8 bytes long). Read from the file using system I/O (i.e. generate an exception using the *svc* instruction). Process the input values one at a time using a loop (be sure to detect end-of-file correctly), calculate e^x and e^{-x}, and then use printf() to print out the input value and its corresponding output values in table form (i.e. in columns, with column headings) to the screen (standard output). Print out all values with a precision of 10 decimal digits to the right of the decimal point.

You can compute the functions e^x and e^{-x} according to the following series expansions:

$$e^x = 1 + \frac{x^1}{1!} + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

$$e^{-x} = 1 - \frac{x^1}{1!} + \frac{x^2}{2!} - \frac{x^3}{3!} + \frac{x^4}{4!} + \ldots$$

where x is a real number input to the functions. Continue to accumulate terms in the series until the absolute value of the term is less than 1.0e-10.

Run your program using the input binary file supplied on D2L. Capture its execution using the *script* UNIX command, and name the output file *script.txt*.

New Skills need for this Assignment:

- Use of system I/O (exceptions) to open and read an input binary file
- Understanding and use of floating-point single and double formats
- Use of floating-point instructions to do simple calculations
- Use of floating-point comparison instructions

Submit the following:

1. Your assembly source code and script via electronic submission. Use the *Assignment 6* Dropbox Folder in D2L to submit electronically. Your TA will assemble and run your program to test it. Name your program *a6.asm* and the script as *script.txt*.

Computing Machinery I
Assignment 6 Grading

Student:_____

Command line arguments	2	_____	
Loop to read in data	2	_____	
File I/O using exceptions	4	_____	
e^{x} routine	4	_____	
e^{-x} routine	4	_____	
Screen output using printf()	2	_____	
Correct use of floating-point instructions	2	_____	
Script showing I/O	2	_____	
Complete documentation and commenting	4	_____	
Design quality	2	_____	
Total	**28**	**_____**	**_____%**