CSE 2312: Programming Assignment 3

Write assembly language functions that implement the following C functions:

1. void sortDecendingInPlace (int32_t x[], uint32_t count)

// input: array (x) containing count entries
// output: array (x), with the values sorted in descending order

2. float sumF32(float x[], uint32_t count)

// returns the sum of the elements in an array (x) containing count entries

3. double prodF64(double x[], uint32_t count)

// returns the product of the elements in an array (x) containing count entries

4. double dotpF64(double x[], double y[], uint32_t count)

// returns the dot product of two arrays (x and y) containing count entries

5. float maxF32(float x[], uint32_t count)

// returns the maximum value in the array (x) containing count entries

6. double absSumF64 (double x[], uint32_t count)

// input: array (x) containing count double entries
// output: the absolute value of the sum of the elements in array (x) containing count entries

7. double sqrtSumF64(double x[], uint32_t count)

// input: array (x) containing count double entries
// output: the square root of the sum of the elements in array (x) containing count entries

8. double geoMean(double x[], uint32_t count)

// input: array (x) containing count double entries
// output: the geometric mean of the elements in array (x) containing count entries

9. char getDirection (BUSINESS business[], uint32_t index)

// input: array of BUSINESS structs and index of BUSINESS
// output: street direction of BUSINESS (e.g., ‘N’, ‘S’, ‘E’, ‘W’)

10. uint32_t getAddNo (BUSINESS business[], uint32_t index)

// input: array of BUSINESS structs and index of BUSINESS

// output: address number of business

11. char * getCity(BUSINESS business[], uint32_t index)

// input: array of BUSINESS structs and index of BUSINESS
// output: city name of BUSINESS

BUSINESS will be provided in the sample driver. For a list of VFP instructions, go here: http://www.keil.com/support/man/docs/armasm/armasm_dom1361289934045.htm

All of the functions above should be present in a single .s file. The function/procedure names must be identical to that presented above, as your code will be tested with generic C code used by the TAs.

Submit your assignment via the submission link on Canvas. The name of this file should be **lab3_lastname_loginID.s**. Example: If your name is John Doe and your login ID is jxd1234, your submission file name must be “lab3_Doe_jxd1234.s”.