Assignment #5 – Simple Functions **CS 218 Fall 2021**

Purpose: This program will involve writing simple functions to perform a variety of tasks.

You will need to write this program from scratch.

Program Specifications

Write a program that will repeatedly prompt a user to input integer numbers until they enter "Quit". You must use the functions below to prompt the user, check their response, and convert to an integer. Output "Valid Numeric Value" if their number is accepted.

You will need to write the following functions:

Get Null Terminated String Length

Returns the length of the string (not including the null) as a 32 bit unsigned integer.

Argument 1: Address to a null terminated string

Print Null Terminated String

Use the string length function (above) to determine the length of the string and use the write system service to print the string to the screen. This function has no return value.

Argument 1: Address to a null terminated string

Prompt User

This function will use the print system service (above) to output a request to the user and read in the user's response using a read system service. Check that the data returned from the user fits within the specified buffer. Replace the linefeed with a null character. If it does not fit within the buffer, clear any remaining input and return -1. If it does, return the size of the input (including the null).

Argument 1: Address to a string prompt to output
Argument 2: Address to a string buffer
Argument 3: Maximum input size (32 bit unsigned int)

Use a buffer of size 100 bytes for this program.

Compare Strings
Compare the strings character by character. If the first string is greater, return 1. If the second string is greater, return -1. Otherwise, return 0 if they are equal.

Argument 1: Address to null terminated string
Argument 2: Address to null terminated string

Convert Integer Format String to 32 Bit Integer
Use the algorithm from assignment #4 to write a function to convert an integer format string to its signed 32 bit integer equivalent. In addition, the function must do some error checking to ensure that the string is correctly formatted.

In addition, spaces should be allowed before the sign or after the final digit in the string.

Output an appropriate message if an error occurs. These error messages should be stored as global variables.

Return a 1 if the number is converted successfully, or -1 otherwise.

Argument 1: Address to a null terminated string
Argument 2: Address to a 32 bit location to store the converted value

Check for the following:

Number below minimum: –2,147,483,648
Number above maximum: 2,147,483,647
Unexpected character found including internal spaces
At least 1 numeric digit

Example bad inputs:

“- 41”	Unexpected Character Space after sign
“ +5k”	Unexpected Character ‘k’
“+”	No Digits
“4000000000”	Number Above Maximum Limit

Submission

Once you are satisfied with the program, upload the assembly source code (.asm) file to the class website.