

**Purpose:** This program will involve writing 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 KMB format 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 KMB Number" if their number is accepted.

You will need to write the following functions:

**Get Null Terminated String Length**

Returns the length of the string (including the null).

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 call to print it 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 call. Check that the data returned from the user fits within the specified buffer. Replace the linefeed with a null character. If it does not, clear the buffer and return -1. If it does, return the size of the input (including the null).

Argument 1: Address to string to output

Argument 2: Address to string buffer to store response

Argument 3: Maximum accepted input size

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 equal.

Argument 1: Address to null terminated string

Argument 2: Address to null terminated string

**Convert KMB Format String to 64 Bit Integer**

Use the algorithm from assignment #4 to write a function to convert a string in KMB format to its signed integer equivalent. In addition, the function must do some error checking to ensure that the string is correctly formatted. Output an appropriate message if an error occurs. These messages should be stored as global variables.

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

Argument 1: Address to null terminated string

Argument 2: Address to 64 bit location to store the converted value

Check for the following:

Number below minimum: -999,999,999,999

Number above maximum: 999,999,999,999

Unexpected character found including non-leading spaces

Example bad inputs:

"- 41k"	" +9.6m"
"b2"	"-1.2k "
"-4.1.6k"	"-16 .3k"
"-5.b"	"7.77 B"
"12g"	".63001m"

**Submission**

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