CMPS 351: Spring 2013 Program #6
Date assigned: Thursday, February 28, 2013 Due: Thursday, March 7, 2013

This assignment asks you to write a MIPS function to reverse the characters in a string. Your function should take two parameters—the address of the string and its length—and use them to create a new string which is the reverse of the given string. Strings in MIPS and many languages like C are slightly different from basic arrays of integers. They are defined as an array of characters, where each character is represented by an 8-bit (1-byte) unsigned number and the last element in the array is 0. To manipulate the memory properly, you will need to use lbu (load byte unsigned) instead of lw to read each character and sb (store byte) instead of sw to save each character. You will also need to be sure to allocate room for one more character than the length parameter would suggest so that you can terminate your string with 0. Note that your function should not try to put the terminating 0 at the front of the reversed string. If your function has been provided with bad parameters, then it should return a null pointer (i.e. $v0 should be 0.)

Your program will not need to interact with the console. Instead you will use a test suite that has been provided for you. The test suite will call your function (String_Reversal) with the parameters ($a0 contains the address of the string, $a1 is the length of the string) and will wait for your function to return its results in $v0. The test suite will also tell you whether your function has correctly calculated the results. Finally, the test suite will also test to make sure that you are following assembly language conventions. This test suite is more extensive than the previous one and will check to make sure that your code can handle bad data. Tests 6-10 provide your function with a set of parameters that cannot be used. Your function should check for these cases before processing the string itself, otherwise, your function will probably crash. For tests #9 and #10, remember that in the 32-bit MIPS architecture, the memory runs to 0x7FFF FFFF and attempting to access memory beyond that point will crash the program.

You should attach your code to the test suite for testing by running either:
Windows: copy /Y <String Reversal Function Name>.asm + "Program #6 - Test Suite.asm" <output>.asm
Unix: cat <String Reversal Function Name>.asm "Program #6 - Test Suite.asm" > <output>.asm
The result of either of these commands is a file that contains both sets of code, so that SPIM can load all of it at once. Your final submission should only include your function and not the test suite.

Your program should include appropriate comments indicating what the code should be doing and which registers are being used. Please include your name and CLID in the program headers and include your CLID in the file names. Your programs should be turned in through Moodle before class starts on the due date. You should test your programs using the SPIM simulator before submitting them.

Expected output:

```
Test #1 passed.
Test #2 passed.
Test #3 passed.
Test #4 passed.
Test #5 passed.
Test #6 passed.
Test #7 passed.
Test #8 passed.
Test #9 passed.
Test #10 passed.
```

CMPS 351: Spring 2013 Program #6
Date assigned: Thursday, February 28, 2013 Due: Thursday, March 7, 2013

Objectives:

1. To review MIPS functions.
2. To practice manipulating arrays in MIPS assembly language.
3. To introduce and practice manipulating the heap in the MIPS architecture.
4. To introduce and practice manipulating ASCII characters in the MIPS assembly language.
5. To practice using a test suite.

Point Values:
Total. 100pts