

11.2 Lab Assignment 8

Start Assignment

Due May 8 by 11:59pm **Points** 95 **Submitting** a file upload **File Types** asm and s
Available after Apr 26 at 10am

Purpose: Write a program and a function where the function needs more than 4 parameters, and utilizes the stack.

Background Of MIPS functions and using the stack

MIPS Functions

MIPS functions have special registers for handling parameters passed to a function when a function call occurs. MIPS also has special purpose registers for handling values needed between function calls:

- registers **\$a0**, **\$a1**, **\$a2**, and **\$a3** are the *4 parameter registers* that a MIPS function uses to pass values into a MIPS function
- registers **\$s0**, **\$s1**, **\$s2**, **\$s3**, **\$s4**, **\$s5**, **\$s6**, and **\$s7** are used to hold values that are meant to be preserved across functions calls;
they should hold values that remain the same no matter what function is being executed

This means that a MIPS function only has register hardware to handle no more than 4 function parameters. If a function ever has more than 4 parameters passed to it, the MIPS stack and other registers must be used to allow more than 4 parameters to be passed to the function.

Stack

Arranging data using a stack occurs often in computer science. Data arranged using a stack can take on interesting properties. One property is that data pushed on a stack is popped off the stack in the reverse order of how it was pushed.

The following six external website reading pages shows an example of how to use the MIPS stack to reverse a string (ignore the question and answer sections on these pages):

[Reverse A String With MIPS Stack Page 1](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_10.html) [_ \(https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_10.html\)](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_10.html)

[Reverse A String With MIPS Stack Page 2](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_11.html) [_ \(https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_11.html\)](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_11.html)

[Reverse A String With MIPS Stack Page 3](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_12.html) [_ \(https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_12.html\)](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_12.html)

[Reverse A String With MIPS Stack Page 4](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_13.html) [_ \(https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_13.html\)](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_13.html)

[Reverse A String With MIPS Stack Page 5](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_14.html) [_ \(https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_14.html\)](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_14.html)

[Reverse A String With MIPS Stack Page 6](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_15.html) [_ \(https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_15.html\)](https://chortle.ccsu.edu/AssemblyTutorial/Chapter-25/ass25_15.html)

The MIPS stack is also used to process function calls and any values that need to be placed into registers for those function calls.

Program

Write a program that takes 5 characters as parameters to a function named **reverse5**.

You are to write your program with a main label that has programming logic to call reverse5.

You should write reverse5 to take 5 parameters.

Use these characters in the part of the program with the main label:

H, E, L, L, O

These 5 characters are the 5 characters to pass to reverse5. They are to be arranged so that:

“H” is the first character parameter,

“E” is the second character parameter.

“L” is the third character parameter,

“L” is the fourth character parameter.

“O” is the fifth character parameter.

Each parameter must be a single character. None of the parameters are allowed to be **strings**. : *sequence of characters*

reverse5 should do the following:

1. pushes H on the stack first
2. pushes E on the stack second
3. pushes L on the stack third
4. pushes L on the stack forth
5. pushes O on the stack fifth (last)
6. prints each character before it is popped off of the stack
7. pops the characters off the stack in the reverse order that the characters are placed on the stack. This means the characters should be printed and the stack popped in this order:
 1. print O then pop
 2. print L then pop
 3. print L then pop
 4. print E then pop
 5. print H then pop for the last time

This means when reverse5 has returned to where it was called, the printing should output the characters to display in this way: OLLEH

Note: Sample Program 2 given below demonstrates the use of programming with a main label and a function that reverses character parameters. Use this program and its programming logic as the basis for your program.

[Use The MIPS Technical Document To See The MIPS Assembly Language instructions](https://ccsf.instructure.com/courses/47907/files/7405493/download?download_frd=1) ↓

(https://ccsf.instructure.com/courses/47907/files/7405493/download?download_frd=1) . You may wish to download the document and refer to it as you are writing the program.

Write comments in your program that state your name, the programming logic, and any details you feel you'd like to explain that state about how you are using the assembly language instructions.

This program is to be saved in its own file, with a .asm or a .s file extension. You may only use a .asm or .s file extension (MIPS assembly language programs are named using these files extensions). Once your program is working correctly, submit the program file to this assignment to receive a grade for your program.

Example Programs

Below is the source code of some example MIPS assembly language programs that can be used to help you write your programs.

Sample Program 1: This program is a version of the string reversing program described in the six external website reading pages from the Stack section above [stringstack.asm](#) ↓ (https://ccsf.instructure.com/courses/47907/files/7748194/download?download_frd=1)

Sample Program 2: This program uses a function that work similarly to the reverse5 function you are to write. It uses only 4 parameters [characterstack.asm](#) ↓ (https://ccsf.instructure.com/courses/47907/files/7748195/download?download_frd=1)