

CS 252 Fall '18

Homework 8

(40 points)

Instructor: Prof. Adil Ibrahim

Due Date: December 5, 2018

Primary Contact for this homework: Amogh (asjoshi4@wisc.edu)

Program 1 (20 points)

Write an assembly program to right shift a given number by 4 bits for a positive 2's complement number.

For example, right shift by 4 bits on a binary string, say **00111111**, will shift bits to right, appending 0 to the left at every shift (0000**0011**).

Your program should –

1. Read a 16-bit positive value stored at memory location Loc1.
2. Store the 16-bit result in Loc2. You can declare the memory locations Loc1 and Loc2 using the ".BLKW" directive. For example, if [Loc1] = 0x00A0 then after execution of the program [Loc2] = 0x000A.
3. Your code should start at memory location 0x3000.

(**Hint:** Right shift operation can be obtained if you divide the number by 2. This implementation is to be done only for positive numbers. You can create a subroutine DivideBy2 and call it 4 times to shift by 4 bits. You can come up with other methods as well but make sure to use **subroutines**).

Program 2 (20 points)

Write an assembly program to reverse a sentence and store the reversed string **in place** (In place reversal implies that the original string is replaced with the reversed string at its location).

Also, display the reversed string on the screen.

The string should start at memory location 0x5000 and your code should start at memory location 0x3000).

For example, if the original starting at 0x5000 was "Awesome!", the reversed string should be "!emosewA" occupying the same location from 0x5000.

Note: Your code will be graded through automated scripts. Name your files as **q1.asm** and **q2.asm** respectively. You must upload a .zip file named HW8.zip which has both Q1.asm and Q2.asm.

To check the correctness of your code, you may run the given sample scripts. Your code will be tested on different test cases. Run "script q1.lcs" and "script q2.lcs" to test the correctness.

TRAP CODES

<i>Code</i>	<i>Equivalent</i>	<i>Description</i>
HALT	TRAP x25	Halt execution and print message to console.
IN	TRAP x23	Print prompt on console, read (and echo) one character from keybd. Character stored in R0[7:0].
OUT	TRAP x21	Write one character (in R0[7:0]) to console.
GETC	TRAP x20	Read one character from keyboard. Character stored in R0[7:0].
PUTS	TRAP x22	Write null-terminated string to console. Address of string is in R0.

ASSEMBLER DIRECTIVES

<i>Opcode</i>	<i>Operand</i>	<i>Meaning</i>
.ORIG	address	starting address of program
.END		end of program
.BLKW	n	allocate n words of storage
.FILL	n	allocate one word, initialize with value n
.STRINGZ	n-character string	allocate n+1 locations, initialize w/characters and null terminator

