Homework 8 [Due at lecture on Fri, May 1]

Primary contact for this homework on Piazza is Lisa Ossian

Important Notes:

- Answers to questions 1, 2 and 3 should be handed to your instructor in hard copy on the due date during the class time.

- Problems 4 and 5 should be submitted electronically to learn@UW. Submission guidelines for Problems 4 and 5:
  
  - If you have first name Jane and last name Doe, turn in one zip file Doe_Jane.zip into the dropbox folder named “homework8” by 1:59PM on Friday, May 1.

  - This zip file should contain the folder Doe_Jane with files hw8_p4.txt and hw8_p5.txt in the folder. These .txt files should contain the assembly code for problems 4 and 5 respectively.

  - Since your code will be tested automatically, it is important to stick to this naming convention, otherwise you will lose credit, even if your code works correctly.

- The programs which you write should always start at address x3000 and end with a HALT instruction (HALT).

- You can submit your code for problem 4 and 5 (ie, the .zip file) as many times as you want, until 1:59 PM on Friday, May 1, 2015. We will consider your latest submission for grading.
Problem 1 (3 points)

This question is based on the "RFID Inside" topic (RFID_INSIDE)

a. (2 points) What is VeriChip? How does it work?

VeriChip, an RFID tag, is a short-range transponder that relies on the signal from a reader unit for its power supply. When exposed to a varying magnetic field from the reader, the chip powers itself up and repeatedly transmits a 16-digit code that is unique to the tag.

b. (1 point) According to the author, what is the best solution for accurate identification of individuals which can avoid the ethical issues raised by the implanted radio chips?

Use a different technology altogether, like biometric scanners.

Problem 2 (3 points)

Assemble and load the following program in PennSim and describe what the program does in 1-2 sentences.

```assembly
.ORIG x3000
LEA R0, PROMPT
AND R4, R4, # -1
LD R2, NEG_ENTER
PUTS
NEXT TRAP x20
TRAP x21
ADD R4, R4, #1
ADD R5, R0, R2
BRz OUTPUT
BR NEXT
OUTPUT LEA R0, PRINT
TRAP x22
LD R3, ZERO
ADD R0, R3, R4
TRAP x21
DONE HALT

PROMPT .STRINGZ "Enter a string (maximum 8 characters): "
PRINT .STRINGZ "Output: "
NEG_ENTER .FILL -10
ZERO .FILL x30
.END
```

Counts the number of characters input by the user (including the ENT).
Problem 3 (4 points)

For the below code, if the label “ONE” is at address x3000, what are the addresses of the labels TWO, THREE and FOUR?

```
ONE .BLKW x0001       x3000
TWO .FILL x2          x3001
THREE .STRINGZ “three” x3002
FOUR .FILL x4          x3008
```

Problem 4 (10 points)

Write an LC-3 program which takes in two numbers from the user, prints these numbers, multiplies the two numbers together, and then stores the result of the multiplication in the memory location corresponding to label “PRODUCT”. You can assume that both input numbers are less than 10.

For example, if the user enters 5 and 3, your program should print the following:

```
5
3
```

Note:
- You need to submit the program as hw8_p4.txt
- Use this template for writing your code: hw8_p4.txt

```
.ORIG x3000

;>>>>>>>>>>>>>INSERT YOUR CODE BELOW>>>>>>>>>>>>>>>>>

; Get first input number
GETC
OUT
LD R5, ASCII2DEC
NOT R5, R5
ADD R6, R5, 1
ADD R3, R0, R6
ST R3, INPUT1
```
; Print new line
LD R0, ENTER
OUT

; Get second input number
GETC
OUT
ADD R4, R0, R6
ST R4, INPUT2

; Multiply numbers together
AND R5, R5, 0
MLOOP ADD R5, R5, R3
ADD R4, R4, -1
BRp MLOOP
ST R5, PRODUCT

LAST HALT

;DATA REGION
;>>>>>>>>INSERT ANY DATA YOU NEED HERE>>>>>>>>>

INPUT1 .BLKW 1
INPUT2 .BLKW 1
PRODUCT .BLKW 1
ASCII2DEC .FILL x30
ENTER .FILL 10

.END
Problem 5 (10 points)

Write a program to take a sentence from the user (not exceeding 30 characters) and reverse the words in the sentence and display the reversed string. For example, if the user inputs as “This is the Last One”, the output should be displayed as “One Last the is This”. Assume that the user terminates his/her input with an enter key and no punctuations are used except spaces. Also assume that the first character is not a space and all words are separated by a single space (no multiple spaces in the input).

Note:
- You need to submit the program as hw8_p5.txt
- Use this template for writing your code: hw8_p5.txt

.ORIG x3000
    LEA R0, MSG
    TRAP x22 ; Puts -- Print the message to user to enter the string
    LEA R1, FCHAR
    ADD R3, R1, 0 ; R3 pointer to FCHAR
    LD R2, ENT ; EOL comparator
    LD R5, NEGSIZE
    NEXT TRAP x20 ; GetC read the char from user
        TRAP x21 ; Out echo it back to the user
    ADD R4, R0, R2
    BRz OUTPUT ; End of input get into output mode
    STR R0, R3, 0
    ADD R3, R3, 1
    BRnzp NEXT

; Data loaded into FCHAR
    OUTPUT ADD R3, R3, -1
    NOT R1, R1
    ADD R1, R1, 1
    LEA R0, PRINT
    TRAP x22 ; Puts caption for output

LOOP ADD R4, R1, R3
BRz DONE

LDR R0, R3, 0 ; LOAD VALUE POINTED BY R3 (POINTER) TO R0
ADD R6, R0, R5 ; CHECK IF R0 IS A SPACE IF SO PRINT THE

RESULTS SO FAR
BRz REVERSE

; TRAP x21 ; OUT the reversed string
ADD R3, R3, -1
BRnzp LOOP

REVERSE AND R0, R0, 0 ; REPLACE RO WITH 0
STR R0, R3, 0 ; REPLACE 0 WITH SPACE
ADD R3, R3, 1 ; INCREMENT R3 TO POINT TO THE NEXT MEMORY

LOCATION TO PRINT DATA FROM THERE.
ADD R0, R3, 0
PUTS
LD R0, SPACE
OUT
ADD R3, R3, -2 ; Iterate repeatedly
BRnzp LOOP

DONE ADD R0, R3, 0
PUTS
HALT
; DATA REGION

MSG .STRINGZ "Please enter a string (max length 30): "
PRINT .STRINGZ "Output:"
ENT .FILL -10 ; new line feed
NEGSPACE .FILL xFFE0 ; decimal -32, -20 - NEGXHEX for space
SPACE .FILL x20
FCHAR .BLKW 31 ; Used for storing the input results from user
.END