

ECE 220 Midterm 1 Practice Questions

Programming:

Part A – see coding questions in Practice_Exam1 and Practice_Exam2.

Part B

- 1 The subroutine ADDPRINT takes two inputs, adds those inputs, prints the sum to the terminal and returns the result of sum in R5. For simplicity, the range of the result will be 0 to 9 only. The following ADDPRINT subroutine however does not behave as specified.

```
1. .ORIG x3000
2.     AND R1, R1, #0
3.     AND R2, R2, #0
4.     ADD R1, R1, #1
5.     ADD R2, R2, #3
6.     JSR ADDPRINT
7.     HALT
8. ;input R1, R2
9. ;output R5, the sum of two inputs
10.    ADDPRINT
11.        ADD R0, R1, R2
12.        LD R3, ASCII_0
13.        ADD R0, R0, R3
14.        OUT
15.        ADD R5, R0, #0
16.        RET
17.
18.    ASCII_0 .FILL x0030
19. .END
```

- 1.A Describe what is happening and find the bug(s) in the given code (provide specific line number).

Your Answer (no more than 30 words):

1.B How would you fix the bug(s)?

Your Answer (no more than 30 words):

2.

In this problem, you will help complete an LC-3 assembly program to remove all spaces in a character string. For example, if you are given a string "Hello World!", the program will convert the string to "HelloWorld!". Here, "_" indicates the space character (ASCII 32). The string is terminated by a NULL character (ASCII 0) and is stored in memory at the memory location indicated by the symbol STRING.

The Algorithm works as follows: We will keep two memory addresses to track the string. One is called "Current Read" address, which is stored in R0. The other is called "Current Write" address, which is stored in R1. In the beginning, both R0 and R1 will contain the starting address of the string. R4 will contain the value -32, which we will use in our comparison tests to check for the space character.

At each iteration, we read the string at the "Current Read" location and test for the space character. If the character is a space, we only need to advance the "Current Read". If the character is not a space, we write the character to the "Current Write" location, and advance both the "Current Read" and "Current Write" locations. We then test for the end of the string. If the character is a NULL, we are done. If it is not, we start another iteration.

Complete the program by filling in the blanks.

.ORIG x3000

LEA R0, STRING ; R0 contains "Current Read" location

ADD R1, R0, #0 ; R1 contains "Current Write" location

____ R4, SPACE ; R4 contains -32 (minus ASCII for space)

NEXT LDR R2, R0, #0 ; R2 contains current character

ADD R3, R2, R4 ; R3 is a temporary value

BR__ NOTSPACE

ADD R0, R0, ____ ; We have a space

BR NEXT

NOTSPACE STR ____, R1, ____ ; Write to "Current Write" location

ADD ____, R0, #1

ADD R1, R1, #1

____ ; Test for end of string

BR__ NEXT

DONE HALT

SPACE .FILL xFFE0 ; -32 in decimal

STRING .STRINGZ "ECE 220 !"

.END

Concepts:

1. Assuming 3 items have been pushed onto the stack. After a POP operation, the last item pushed onto the stack will be erased from memory. (TRUE or FALSE; use no more than 20 words to explain your choice)
2. Polling I/O is more efficient than interrupt-driven I/O. (TRUE or FALSE; use no more than 20 words to explain your choice)
3. In LC-3, what is the benefit of using a subroutine? (use no more than 20 words)
4. Explain what is a stack underflow. (use no more than 20 words)
5. The input stream of a stack is a list of all the elements we pushed onto the stack, in the order that we pushed them. If the input stream is ZYXWVUTSR, create a sequence of pushes and pops such that the output stream is YXVUWZSRT.