

Introduction to Computing: Homework 6

Fall 2025

Dr. Ramesh Yerraballi

The University of Texas at Austin

Due: [Tuesday, November 11th, 11:59pm](#)

Instructions:

You may discuss the problem set solutions with your fellow classmates but the write up must be your own. Please use the TAs and the Instructor for help before you seek out a friend or classmate. Show your work. You may handwrite or type-write your answers.

Note: You are highly encouraged to work out all the problems without using the LC3-simulator. It will increase your proficiency in analyzing code on paper.

1. **(10pts)** *Code Tracing*

A current TA stole this question from the previous TA but he doesn't understand how it works or what it does because of its funny implementation. The documentation only tells him to put a positive value at the label **input** before running the code. Figure out what this program accomplishes.

```

                                .ORIG x3000
                                AND R2, R2, #0
                                .FILL x2008
                                .FILL x5260
                                .FILL x127F
                                .FILL x1402
                                .FILL x1001
                                .FILL x0BFD
                                .FILL x2003
                                .FILL x7400
                                .FILL xF025
input                            .FILL x0008
                                .FILL x3010
                                .END
```

2. (10 pts) *Symbol Tables and Memory*

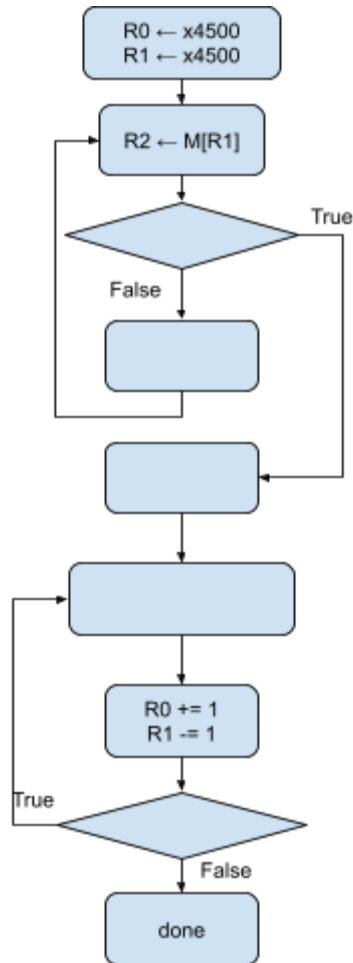
- Write down the Symbol Table generated when the following program is assembled.
- Write down the memory contents for locations x400A to x4014 **before AND after** executing the code

```
.ORIG x4000
AND R1, R1, #0
LEA R2, Arr
Loop BRn End
LD R3, Ct
STR R3, R2, #0
ADD R2, R2, #1
ADD R3, R3, #-1
ST R3, Ct
BRnzp Loop
End HALT
Arr .BLKW 10
Ct .FILL x0009
```

3. (15 pts) *Flowcharts and Arrays*

Given a null-terminated array starting at x4500, Given the following flowchart, fill in the blank to reverse the array.

	Before	After
x4500	x59	x26
x4501	x84	x01
x4502	x40	x40
x4503	x01	x84
x4504	x26	x59
x4505	x00	x00



4. (20 pts) *Writing Code from a problem statement*

Write code to convert a null-terminated octal number in ascii beginning at x4000 to decimal value, and store the result in x5000. Assume the input will be at most 4 digits. *Please limit your solution to at most 30 instructions.*

e.g. given the following memory layout, x5000 would contain 122, or x7A, because the ascii string is "172" and $(172)_8$ is equivalent to $(122)_{10}$

x4000	x31 '1'
x4001	x37 '7'
x4002	x32 '2'
x4003	x00 '\0'
x5000	x7A

hint: think back to how you convert octal to binary, the solution can be less than 20 lines without counting the constants for the addresses, so don't overthink it.

5. (10 pts) Fill in the Blanks

The following code calculates the number of spaces in a null-terminated string of characters starting at x6000 and stores the result at x4500. Fill in the blanks in the code.

```

        .ORIG x3000
        LD R0, StringAddr
        LD R1, Space
        AND R5, R5, #0
        _____ ; Blank 1
Loop     ADD R1, R1, #1
        _____ ; Blank 2
        BRz Finish
        ADD R3, R2, R1
        _____ ; Blank 3
Misc     _____ ; Blank 4
Misc2    ADD R0, R0, #1
        BRnzp Loop
Finish   STI R5, NoOfSpaces
        HALT

StringAddr .FILL x6000
NoOfSpaces _____ ; Blank 5
Space     .FILL x0020
```

6. (10 pts) Code Tracing

What does the following program do assuming that the value stored at x4500 is always positive? (Explain in less than 15 words)

```

        .ORIG x3000
        AND R2, R2, #0
        ADD R2, R2, #1
        ADD R3, R2, #0
        LDI R1, InputAddr
        ADD R1, R1, #-2
LoopityLoop BRnz Done
        ADD R4, R3, #0
        ADD R3, R3, R2
        ADD R2, R4, #0
        ADD R1, R1, #-1
        BRnzp LoopityLoop
Done     ST R3, Result
        HALT
```

```

InputAddr    .FILL x4500
Result       .BLKW 1
              .END

```

7. **(10 pts)** Arrays vs. Linked-Lists - Interpret the following structure as both an array and a linked list.

As an array, the number in the .FILL after the .STRINGZ represents their grade in the class.
 As a linked list, the number in the .FILL after the .STRINGZ represents the pointer to the next item in the linked list. The linked list is sorted from highest to lowest grade. In both ways of interpretation, the structure starts being read at x0000.

```

.ORIG x0000
.STRINGZ "ABE"
.FILL #20
.STRINGZ "BRK"
.FILL #30
.STRINGZ "CAG"
.FILL #25
.STRINGZ "DRK"
.FILL #35
.STRINGZ "ELL"
.FILL #10
.STRINGZ "FLS"
.FILL #5
.STRINGZ "GFY"
.FILL #15
.STRINGZ "HRY"
.FILL #0
.END

```

List the names in order of descending rank. That means the highest grade comes first.

Array Interpretation	Linked-list Interpretation

--	--

8. (15 pts) *Miscellaneous*

Does this program ever terminate? If it does, write the value stored at **Result** after execution and explain. If not, explain your reasoning.

```
                .ORIG x3000
                AND R1, R1, #0
Loop            ADD R1, R1, #2
                ST R1, #-2
                ADD R1, R1, #3
                BRnzp Loop
                ST R1, Result
                HALT
Result         .BLKW #1
                .END
```