Activity: Assembler Programs

A. Why?

- Assembler language is easier to read/write than machine language.

B. Outcomes

By the end of the class you should

- Understand how the LC-3 assembler produces a symbol table.
- Understand the difference between HALT and END in LC-3 assembler.
- Understand the difference between FILL and BLKW in LC-3 assembler.

C. Questions

1. Where will each instruction/fill/etc. be stored at in memory for the program below?

```
; Print a message string that begins with a line feed
;
        .ORIG x3000             ; Start the program at x3000
        LEA     R2, MSG         ; Pt R2 -> start of message string
LOOP1   LDR     R0, R2, #0      ; R0 = next char of string
        BRZ     DONE            ; Loop until end of string
        TRAP    x21             ; Print current char of string
        ADD     R2, R2, #1      ; Pt R2 to next char of string
        BR      LOOP1           ; Continue loop
DONE    TRAP    x25             ; Halt execution
MSG     .FILL   x000A           ; Line feed
        .FILL   x0048           ; "H"
        .FILL   x0069           ; "i"
        .FILL   x0020           ; " "
        .FILL   x0021           ; "!"
        .FILL   x0000           ; end of string
        .END                    ; End of program
```

2. What addresses will the labels from the previous program have? When we first encounter the `BRZ DONE` instruction, do we know where in memory the label `DONE` is defined?
3. What is the difference between the HALT instruction (`TRAP x25`) and the `.END` pseudoinstruction? How many halt instructions can you have per program? How many `.END`s?
4. Suppose we changed `MSG .FILL xA` to `MSG .BLKW xA`; what would be different about our program?