

## Activity: LC-3 Programming

### A. Why?

Practice makes perfect.

### B. Outcomes

By the end of the activity you should

- Know how to translate low-level pseudocode into binary.
- Know how to detect the leftmost bit of a word
- Know how to left-shift a word by one bit.
- Have thought about how to look at the leftmost byte of an LC-3 word.

### C. Questions

1. Translate the program from today's notes into binary.

```

TOP  LDR  R0, R2    ; R0 = M[R2], the current character
      BRZ  DONE    ; (next+3) Done if we find null char
      TRAP x21     ; print char in R0
      ADD  R2, R2+1 ; Pt R2 -> next char
      BR   TOP     ; (next-5) continue loop
DONE  ...

```

2. How can you test the leftmost bit of a word to see if it's 0 or 1? Sketch some code that adds 1 to R1 iff the leftmost bit of R0 is 1.
3. How can you shift the bits of R0 leftward one position, filling in the rightmost bit with a zero? (E.g., shifting the 4 bits 1101 leftward yields 1010.) Hint: What do you need to multiply the bitstring by? How do you do that multiply?
4. Using just the right byte of a word to hold a character is kind of wasteful; it would be more space-efficient to use both bytes. Discuss in group: How could you take the left byte of R0 and make it the right byte of R1? (Don't actually try writing out the code; it's nontrivial.)