Lab 5
More loops, and reading code

20 February 2025

The preview for this lab is given below. Come to lab on Thursday either with it completed or with a specific written question in your notebook identifying which preview step you got to and what about it you're stuck on.

Part 1a: finding the earliest letter

For the first part this week, you'll write a program that reads a single word and prints out the alphabetically-earliest letter in that word. You can assume that the word is letters only and will be either all-caps or all-lowercase (but it could be either one!).

1. Open the codeboard assignment for Lab 5 part 1a. Enter the boilerplate code (this should be feeling a bit more automatic, but you can still look it up if you need to!)
2. Add code to read in a single word. Don't prompt the user for it (it'll break my test cases).
3. Write a loop that accesses each character of the word.
4. (Are you remembering to check after every step that your program compiles and runs, and submit it often?)
5. Add three lines of code to build an accumulator. In this case, the accumulator will be tracking the earliest letter seen so far (which at the end will thus be the earliest letter overall). Remember the three items on the checklist:
   - Declare-and-init the accumulator (before the loop)
   - Update the accumulator (inside the loop)
   - Use (e.g. print) the accumulator (after the loop)

   If you're not yet sure what you should use to init the accumulator, put *some* arbitrary value in there (and fix it later). If you're not sure how

to update, put a comment inside the loop reminding you to update the accumulator (and fix it later).

6. Fix the body of the loop, if you haven't already. When and how is it appropriate to update the earliest-letter-seen-so-far value (which is what we're storing in the accumulator)?

7. Fix the initial value of the accumulator, if you haven't already. What is a "safe" initial value for this variable that won't break the rest of the algorithm? (There are at least two correct answers to this question, in the context of the current problem.)