1. Type in the following list of lists (these are names and ages): \( L := [[\text{Jack}, 35], [\text{Sawyer}, 32], [\text{John}, 59], [\text{Walt}, 16], [\text{Kate}, 30], [\text{Charlie}, 26]]; \)

(a) (1 point) Tack on 4 more names and ages and use \texttt{nops} to see how many entries are in the list (should be 10).

(b) (2 points) Set up a for loop to find John’s entry and print out his age in a nice sentence (with either print or printf), e.g., “John is 59.” The number 59 should not appear in your loop (i.e., you have to grab it from the list once you spot John’s name)!

(c) (1 point) Toss in an extra line after your loop to express your condolences that the name was not found (how will you recognize that it was or wasn’t found?).

(d) (2 points) Set up another for loop to spit out the sentence “NAME is under 30” (where NAME is actually the name, not the word “NAME”) for anybody under 30. ALSO have your loop print out a statement at the end saying “There are NUM people under 30.” (where NUM is, of course, the number).

2. (4 points) Let’s see if we can decide how much time \( \sin(x) \) spends above the \( x \)-axis. Write a while loop to grab random floating point numbers between 0.0 and 50.0 and plug them into \( \sin(x) \). Keep track of both the number above the \( x \)-axis and the total number of numbers you have tried. Terminate the loop once you have found 100 numbers above the \( x \)-axis. Also, at the very end of the loop, spit out a sentence saying that “NUM of NUM trials were above the \( x \)-axis.” (Warning: Save your work before you try running this. It is very easy to have while loops turn into infinite loops. If this happens, try clicking the stop sign at the top of the screen. If that doesn’t work, let me know....)