Administrative Reminders

• Exercise 3 due tonight.
• Lab6 due this Friday as usual.

• Note: I will be pushing A1’s due date *after* the midterm. Will release A1 by next week.
Exercise 3 – Some advice (not limited to E3)

- Before you start writing code, think about the process you’d need to follow to solve it as a person first!

- Any time you find yourself saying: “I should only do X if Y is True”, this is an indication you need an if statement. Possibly followed by elif or else or multiple if-statements depending on what you’re trying to do.

- Any time you find yourself thinking: “I should remember the values I’ve seen so far or something along these lines”, this is an indication you need to have a variable that refers to that values or values.
Some advice (cont’d)

• For every line of code you write, you need to think about what its purpose is.

• Trace through your code, esp. loops, by hand as I had you do in Lab5. Write down the values for every variable at the end of each loop iteration.
  • Ask yourselves: Is this behaviour different than what I’d expect/want? Why? If yes, fix. And repeat.
repeat_letter function

• Break the problem into smaller steps.

• For example, if I gave you a string \( s1 = \text{“hello”} \), and I asked you to create a new string that is the original string with its last character replicated 4 extra times, can you write an expression in your shell that does that?
  • Write the expression in such a way that it is general, i.e., it works for any last character and not just the ‘o’ here.

• Now, can you write an expression that creates a new string where the character at index 2 of string \( s1 \) is replicated 4 times?
sum_numbers function

• You are not permitted to use lists to solve this.
• And yes, this means you are NOT permitted to use string method split either, as it returns a list.

• HINT: Do you recall the different types of accumulators we’ve talked about so far?
  • We have talked about numerical accumulators, string accumulators, etc.