CS1114 Lists Problem Set

October 23, 2023

Instructions:

- Every student, however, must submit his or her own submission for credit.
- Questions do not necessarily appear in order of difficulty
- Each solution should be thoroughly **designed**, following each step of the design recipe in detail.

1. **Composed Image**

   Create a data definition for a composed image. A composed image may contain one or more rectangles and circles that are above, next, or overlaid in relation to each other.

2. **Sum of Reciprocals**

   One somewhat common operation across various areas of mathematics is to compute the sum of the reciprocals of a series of numbers https://en.wikipedia.org/wiki/List_of_sums_of_reciprocals. There are many areas where this comes up. Given a finite series of numbers, your job is compute the sum of the reciprocals of that series. You can assume your inputs are in a fractional form (when $0 < |n| < 1$). Your answer should be in fractional form as well.

3. **Average positive rainfall delta**

   Design a program called average-positive-rain-delta that consumes a list of numbers representing daily change in rainfall amounts (as entered by a user). The list may contain the number -999 indicating the end of the data of interest. Produce the average of the non-negative values in the list up to the first -999 (if it shows up). There may be negative numbers other than -999 in the list. These are considered valid data. Do not neglect the empty list case.

4. **Harvard Bridge Length**

   The Harvard bridge in Boston is 659.82 meters long. In 1958, the bridge was measured using a unit of length called a "Smoot", a unit of length equal to 5 feet 7 inches, the height of a fraternity pledge named Oliver Smoot, https://en.wikipedia.org/wiki/Smoot. We could imagine using other people's heights, though, as units of measurement to measure the length of the bridge. Design a function that consumes a list of heights (X ft, Y in) and produces a list of the lengths of the bridge in each of those units.

5. **The No-shows**

   At a party, there's a list of guests who RSVP'd and another list of guests who actually attended. Design a function that takes in both lists and returns a list of guests who RSVP'd but didn't attend. Given the lists '(Alice Bob Charlie) and '(Alice David), the function should return '(Bob Charlie).

6. **Reorganizing books**

   Your friend is rearranging a pile of his books. He wants to insert a specific book right after a given title. Design a function that takes in a list of book titles, the title of the book to find, and the title of the book to insert. The function should return the updated list of book titles. As an example, given the list '("Moby Dick" "Pride and Prejudice" "Hamlet"), inserting "The Great Gatsby" after "Pride and Prejudice" should have "Moby Dick" then "Pride and Prejudice" then "The Great Gatsby" then "Hamlet". Be sure and think about what should happen if title to search for is not in the list.

7. **A Shopping List**

   A cook has two lists of ingredients: one for a main dish and one for a dessert. Obviously each item has a name and a quantity. He wants to create a single shopping list for the requirements to make both dishes. Design a function that takes the two lists and returns a single list with required items and quantities from both. If the same ingredient is on both lists, do not list it twice on the shopping list; instead, just add it once, but make sure you buy enough of it for both recipies.

8. **Kebab**

   A chef is preparing a shish kebab and wants to remove the topmost piece of meat from it. Design a function that takes in a list representing the kebab's ingredients and returns the

kebab without its first piece of meat. Given a kebab with garlic meat bell-pepper meat onion on it, your function should return a kebab with just garlic bell-pepper meat onion.