# Racket Programming Assignment #4: Recursive List Processing and Higher Order Functions

## **Learning Abstract**

This Racket Programming Assignment #4 is a total of 10 problems. Problems 1-4 use recursive functions that will help in the remaining problems. Problems 5 we can critically exam the code in order to answer 15 different questions. Problems 5-10 will use the first 4 problems as well as higher order functions in order to give me more experience using them. Overall the assignment is excellent practice of lambda functions, recursive functions and higher order functions.

**Problem 1: Count**

**Code:**

**Text

Description automatically generated**

**Demo:**

**Text

Description automatically generated**

**Problem 2: list->set**

**Code:**

**Text, letter

Description automatically generated**

**Demo:**

**A picture containing calendar

Description automatically generated**

**Problem 3: Association List Generator**

**Code:**

**A picture containing text

Description automatically generated**

**Demo:**

**Text

Description automatically generated**

**Problem 4: Assoc**

**Code:**

**Text, letter

Description automatically generated**

**Demo:**

**Graphical user interface, text, application

Description automatically generated**

**Problem 5: Frequency Table**

**Code:**

**Text

Description automatically generated**

**Graphical user interface, text, application

Description automatically generated**

**Demo:**

**Graphical user interface, text, email

Description automatically generated**

**Questions:**

**1. List the names of the functions used within the ft function that you were asked to write in this programming assignment.**

I wrote the count, list->set, and a-list function in order for the ft function to work.

**2. Within the ft function, what function is provided to the higher order function map? Since you cannot name this function, please write down the complete definition of this function.**

( map ( lambda (x) ( count x the-list ) ) the-set )

The higher order function map is using a lambda function. The lambda function being used takes in one parameter, x, which is plugged into the function later as ‘the-set’, defined right beforehand. The-set is defined as the-list being plugged into list->set which removes all multiples in a list to give you a set of one of each item. When the lambda function takes this set in as an argument, it passes it to the count function, which counts the-set inside of the-list.

**3. How many parameters must the functional argument to the application of map in the ft function take?**

2

**4. What would be the challenge involved in writing a named function to take the place of the lambda function within the ft function. Do your best to articulate this challenge in just one sentence.**

The function would have to go through each value inside the list, and then still be mapped properly to the set making it difficult because map only takes in 2 arguments.

**5. Within the ft function, what function is provided to the higher order function sort? Since you can name this function, please simply write down its name.**

Association-list

**6. What is a “keyword argument”?**

A keyword argument is specified with a keyword followed by the argument expression. The argument expression is often used as a key in a map function to find a specific object/item.

**7. Within the ft-visualizer function, what function is provided to the higher order function map? Since you can name this function, please simply write down its name.**

Pair-visualizer

**8. Why was the challenge involved in using a named function in the application of map in the ft function absent in the application of map in the ft-visualization function?**

Because there are only 2 arguments provided for map, in which case the 2nd argument serves as the argument for map and the function.

**9. Within the pair-visualizer function, what function is provided to the higher order function foldr? Since you can name this function, please simply write down its name.**

String-append

**10. Does the add-blanks function make use of any higher order functions?**

no

**11. Why do you think the display function, with the empty string as its argument, was called in the ft-visualizer function?**

This is used to help format the output properly.

**12. What data structure is being used to represent a frequency table in this implementation? Please be as precise as you can be in articulating your answer, preferring abstraction to detail in your precision of expression.**

Stem and leaf structure. The function uses dotted pairs from the ft to output stars in the visualizer.

**13. Is the make-list function used in the pair-visualizer function a primitive function in Racket?**

Yes

**14. What do you think is the most interesting aspect of the given frequency table generating code?**

I enjoy that it uses the functions we made earlier inside the ft function. It gives me a better understanding on how the code works overall.

**15. Please ask a meaningful question about some aspect of the accompanying code. Do your best to make it a question that you think a reasonable number of your classmates will find interesting.**

I believe this code functions very well and is written well. However it may be difficult for a new user to racket to understand. Is there a way you would’ve writen it differently to make it easier to read for beginners?

**Problem 6: Generate List**

**Code:**

**Text, letter

Description automatically generated**

**Demo 1:**

**Text, letter

Description automatically generated**

**Demo 2:**

**Chart

Description automatically generated**

**Demo 3:**

Shape, circle

Description automatically generated

**Problem 7: The Diamond**

**Code:**

**Text

Description automatically generated**

**Demo 1:**

**Icon

Description automatically generated with medium confidence**

**Demo 2:**

A picture containing background pattern

Description automatically generated

**Problem 8: The Diamond**

**Code:**

**Text

Description automatically generated** **Text, letter

Description automatically generated**

**Demo:**

A screenshot of a computer

Description automatically generated with low confidence

**Problem 9: Transformation of a Recursive Sample**

**Code:**

**Text

Description automatically generated**

Text

Description automatically generated

**Demo:**

Text

Description automatically generated

**Problem 10: Blood Pressure Trend Visualizer**

**Code:**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Text

Description automatically generated**

**Demo:**

Text

Description automatically generated

Text

Description automatically generated

Chart, bubble chart

Description automatically generated