**ChatGPT 3.5 Prompts**

(Somewhat embarrassing)

1. Give me a comprehensive overview of how I might implement a recursive-descent parser in Racket
2. How can I create a tokenizer/lexer for Racket in the following grammar? (Fall23Grammar was attached)
3. Is there a function to return a list of all the matching substrings in the input string according to a specified pattern?
4. What does the ‘\\’ mean in front of different operators/characters in the regular expression?
5. (A) Give me an overview of the different types of conditional statements I can use in Racket. (B) Give me an example of how I might wrap the regexp-match\* function in a conditional statement, including some sort of error handling mechanism
6. Give me an overview of the ways I can perform comparisons in Racket
7. Explain how equal? works in more detail. How is it different from eq?
8. How can we iterate through a list without tracking an index? (For ex. i = 0 in C++/Java) Is there a way to return a list with the first element removed? What methods exist in Racket for accessing elements in a list?
9. Explain car, cdr, cddr, and the map function in more detail
10. Write functions for me that check if a string is an id or a num given the following grammar (grammar was attached)
11. Try that again using regex
12. How can I write a regular expression that checks for the optional sign in front of a num?
13. What are maybe/either types in Racket and how do I return one from a function?
14. I have a function isStmt? that's throwing an error for me. (“Not a procedure, expected a procedure that can be applied to arguments.”) I understand what it’s saying, but I just want isStmt? to always return true as a placeholder until I have time to implement it. How can I do this?
15. I guess what I’m asking is if there’s a way I can define a function in Racket that takes arguments but doesn’t actually use them as a placeholder. I think the issue is that Racket doesn’t consider that a procedure.
16. How can I read in a file and split the lines of that file into separate lists of strings, separated by whitespace. Is there a function that exists to do something like this?
17. Is there a more concise way to do this? Elaborate on how I can apply the ‘string-split’ function to the lines using ‘for/list’
18. How can I use the displayln function in Racket to output the concatenation of a string and some custom text, for example something like (displayln string + "This line passed")
19. Here's a function definition for a Line\_List. I'm getting an error here "car: contract violation, expected: pair?, given '()" The input I'm passing to the function is a list of lines from a program split into strings: ((num1 = 12 \* 3;) (num2 = 5 - 2;) (while ((num1 \* / num2) > (num1 / 2))) (x = num1;) (y = num2;) (gosub function;) (num1 = x;) (write num1;) (endwhile;) (write num1;) (write num2;) (end;) (function: x = x + y: y = y \* x: x = x+7: return;) (function: x = 0; return ;) (function: x = 3; end;) ($$)) Here's the function definition itself: (define (isLine-List? input) (or (and (isLine? (car input)) (isLine-List? (cdr input))) (null? input)))
20. Here's a function I'm using to split a string into individual tokens. Later, I call this function with the line "(define tokens (map tokenize input))", where input is a string consisting of a single line. This is working fine, except for when a line ends in something like "2;" for example. How can I get my program to recognize this as a number followed by a semicolon even though there is no whitespace?
21. Is there a simpler way I can do this, since I know the last token in every valid line should end with a semicolon. Can I just check for that semicolon, then remove it from the last substring of the line before I map/tokenize the line into tokens?
22. What if I wanted to put this semicolon check in the isLine? function definition itself and remove the semicolon from the end of the line before calling the tokenize function. Example: (define (isLine? input) ;some code to check for semicolon at end of input and remove (define tokens (map tokenize input)) Assume input is something like (num1 = 12 \* 3;), that is a list of strings
23. This seems to only check the first string in each line (which is a list of strings)
24. How can I check each string in the list for a semicolon character and separate that semicolon into its own element in the list (just after the string I find it in)?
25. What happens if the semicolon is in the middle of a string in the list rather than at the end?
26. Can we do exactly this but keep the semicolons in the list as their own strings?
27. Can you explain the regular expression here in detail?
28. How can I write a regular expression similar to #rx"^(.\*?);(.\*)$" but one which checks for a left parenthesis rather than a semicolon
29. The only issue with this strategy: (define (split-semicolons lst) (define (split-string str) (if (regexp-match #rx"^(.\*?);(.\*)$" str) (list (cadr (regexp-match #rx"^(.\*?);(.\*)$" str)) ";") (list str))) (apply append (map split-string lst))) Is that I seem to lose whatever comes after the character we're splitting the string at. For example, if I have the string ";variable" how can I get ";" and "variable" as elements in the final list
30. Good, one more issue though. If there's a string in the list like "::num" it splits into ";" and ";num" rather than ";" ";" "num"
31. How can I write a function that takes a list of strings and removes any strings that are empty?
32. Suppose I have a list of strings in Racket: ("((num1" "\*" "/" "num2)") I want a function that finds each left parenthesis in each string in the list and separates the string into separate strings at that parenthesis, including a separate string with the single "(" So that the output for this list of strings should be: ("(" "(" "num1" "\*" "/" "num2)")
33. I have a list of strings in Racket that represents a line in a program, which is a list of lists of strings. I pass this line to a function which maps the strings to a list of tokens. I'm wondering if there's a way I can pass a number with the line that represents the place of the list of strings in that larger list of lists.
34. Let's say I have a list of lists. Ex: ((list1) (list2) (list3) (list4)) Is there a way to grab a list's position/index in the list, for ex. 2 in the case of list2
35. How can I make a regex for a label where a label is defined as [a-zA-Z][a-zA-Z0-9]\* followed by a semicolon?
36. I have a list of lists of strings in Racket that looks something like this: (num1 = 12 \* 3;) (num2 = 5 - 2;) (while ((num1 \* / num2) > (num1 / 2))) (x = num1;) (y = num2;) (gosub function;) (num1 = x;) (write num1;) (endwhile;) (write num1;) (write num2;) (end;) (function: x = x + y: y = y \* x: x = x+7: return;) (function: x = 0; return;) My question is how can I search for the string "while" and combine each list of strings until I find the string "endwhile;" (in this example the third list in the list would become (while ((num1 \* / num2) > (num1 / 2)) x = num1; y = num2; gosub function; num1 = x; write num1; endwhile;)
37. string-contains is unbound in Racket
38. What if we did this using the member? function to check if "while" is a member of a list of strings. If it is, append that list together with each list that follows until you find a list with the member "endwhile;"
39. Can you write another function nested inside of this function I have to split sublist2 at the appearance of the symbol 'r-paren. For example: (define (process-list symbols) (let ((result (split-at-bool-match symbols))) (if result (let ((sublist1 (car result)) (sublist2 (cadr result))) ; Process sublist1 and sublist2 separately (display "Sublist 1: ") (display sublist1) (newline) (display "Sublist 2: ") (display sublist2) (newline) (if (and (condition-for-sublist1? sublist1) (NESTFUNCTIONTOSPLITSUBLIST2HERE)) true false)) false)))

**Other Sources:**  
Racket docs, BeautifulRacket, and a little inspiration/impetus from Sailor’s repo for the Spring 2023 class: <https://github.com/Joseph-Sailor-Usher/CS-441-Parser/blob/main/parser.rkt>

(Didn’t look at it that closely, but feel like I’m obligated to mention it since I think it inspired my initial direction with the tokenizer)