Define - Provide example- Understand singificance

3 comprehension questions, out of 4 each

1. difference between linear iterative process and linear recursive process, provide example code function for each

2. explain the difference between lexical and dynamic scoping, and he gives you a code snippet and asks what the output would be using either type

3. define closure, show an example, and explain why it's used in programming

page 2:

1. explain difference between normal and applicative order, then he gives a function and asks for the substitution model for it using each applicative and normal

2. gives you about 5 lambda functions and asks the output

3. gives you a huge list of (cons(cons (cons))) and you need to write the list it creates, and then it asks you caddr, cdaddr, cdaddr and stuff of that list you just wrote

page 3:

1. write a function (define (sum-tree lst)) that takes in a flat, or unflattened list, and returns the summation of all numbers in the list

2. a bunch of code for the next 2 questions

page 4:

2 boxes to do contour diagrams, using the code on the previous page. It was tricky bc the code was an object definition and then calls on the object, so it was unlike any other contour diagram that we had to do and it was fucked lol

page 5:

make a stack object with dispatch for push pop print empty?(checks if empty)

page 6:

-code a split function in prolog which splits a list into 2 lists, the first lists length is passed in by the user

-code an append for prolog

-code a sum 1 to n function in prolog

i think there was something else tbh but i forget

page 7:

given a database of basicparts and assembled parts in this format

basicpart(partname).

assembly(assembled part, [list, of, parts, this, part, requires, for, assembly]).

and you had to code a predicate partof(part, partlist) that takes in any part, and returns a list of all parts it requires for assembly, keeping in mind that a part required might also be another part that requires other parts

==========

1. Difference between Normal and Applicative order. Given code, draw the substitution models for each.
2. Difference between Lexical and Dynamic scoping. Given code, give the the resulting value of x for each
3. What is a closure? Give a code example.
4. What is the difference between a linear recursive process and an iterative process? Give code examples of each.
5. Given some weird nested lambdas (similar to midterm), give the output value.
6. Given a nested list and several statements (car, cdr, caddr, cadadr, etc.), give the output value.
7. Write a procedure that takes a possibly nested list and returns the sum of all elements in that list.
8. Given code (Point object (x, y, getters, setters) using dispatch), draw a contour diagram part way through execution, and another at the end of execution.
9. Implement a stack object with constructor, pop (also returns the element), push, empty?, print using dispatch.
10. Prolog Questions
    a. splitlist(N,L,R1,R2) takes a list, and returns a list with the first N elements (R1) and another list (R2) with the rest of the elements from L
    b. append(L,Z,R) write your own append (in the assignment)

c. sqrlist(L1, L2) checks if all elements in L2 are the squares of all elements in L1
d. nth(N,L,R) takes a list L and an index N and returns the Nth element in L

e. Given the following facts: basicpart(axle), basicpart(fuck), basicpart(shit), assembly(wheel, [axle, fuck]), assembly(thing, [fuck, shit]), write rules that take in a value X and return all basicparts needed to assemble the item (or just X if it's a basicpart).