- 1- Grammar: We handle it in lang.rkt
- 2- Environment: We handle it in environments.rkt
- 3- Data-Structures: We handle it in data-structures.rkt
- 4- Interpreter: We handle it in interp.rkt
- 5- Lexical-Analyzer: We handle it in lang.rkt

B)

[] =>
$$[z = \lceil 3 \rceil]$$
 =>
 $[y = \lceil 2 \rceil, z = \lceil 3 \rceil]$ =>
 $[x = \lceil 1 \rceil, y = \lceil 2 \rceil, z = \lceil 3 \rceil]$ =>
 $[v = \lceil 3 \rceil, x = \lceil 1 \rceil, y = \lceil 2 \rceil, z = \lceil 3 \rceil]$

C)

- 1- Our expressed values are these:
 - num-val
 - bool-val
 - rational-val
 - list-val
 - str-val
- 2- Our denoted values are these:
 - num-val
 - bool-val
 - rational-val
 - list-val

Workload Breakdown:

We answered the first 3 questions as a team by brainstorming. In addition, we did debug together and finished our program.

Mustafa:

- Worked on data-structures.rkt, environments.rkt and interp.rkt.
- Implemented rational numbers and their implementations on op-exp and zero?-exp.
- Added simpl-exp for rational numbers. Wrote its helper function gcd.
- Populated init-env with x y z.

- Tolga:

- Added list-val into the expval struct in data-structures.rkt
- Added expval->list function in data-structures.rkt
- Added list-exp to the grammar and interpreter
- Added cons-exp to the interpreter
- Added sum-exp to the interpreter

- Görkem:

- Added max-exp and its helper function to the interpreter
- Added grammar to the lang.rkt
- Checked the lexical-spec part in lang.rkt