COMP 301 - PROJECT 4

Team members:
Muhammet Eren Ozogul
Ada Yıldız
Aynur Ceren Cepni

Part 1:

- **The code passes all the test cases.**

data-structures.scm:

We added nested-procedure and extend-env-rec-nested.

environments.scm:

We extended the environment with the count variable whose value was 0 and modified apply-env to make it work with nested structures.

interp.scm:

We wrote the cases for value-of for proc-nested-exp, call-nested-exp, and letrec-nested-exp. Also we modified the apply-procedure to print the count of the procedure call.

lang.scm:

We added the grammar for proc-nested-exp, call-nested-exp, and letrec-nested-exp.

translator.scm:

We added translation implementation for proc-exp, call-exp, and letrec-exp. In call-exp we increment count or initialize it to one and return a call-nested-exp.

Part 2:

Translator.scm:

We implemented apply-senv-number, proc-exp, let-exp and var-exp. Apply-senv-number works in a similar way to apply-senv the only difference is that it increments the number when the variable is in our environment to count the occurrences of the variable. And for the implementation of let-exp, var-exp and proc-exp we changed them accordingly to call apply-senv-number to return the count of occurrences of our variables and manipulated the strings with given built-in procedures to print out the exact same outputs given for each test case provided in the tests.scm file.

Workload Distribution:

Part 1: We focused on this question together.
Part 2: We focused on this question together.