1 Week 8

1. Analyse the **parser** for If-then-else statement. The program prints a parse tree. Start with the analysis and compilation of tree.c file.

   Allowed structures:

   ```
   if (a>2) then a=2
   if (3>a) then if (b>=2) then b=2
   if (3>a) then if (b>=2) then b=2 else c=10
   2
   c=10
   ```

2. Write an interpreter for lisp, starting from lisp.l and lisp.y

   a) test_cons file defines a structure for cons cell and functions. Compile the file and run it. You will use the functions and structures from here in lisp.y.

   b) Use lisp.l and lisp.y and replace /* your code here*/ with needed code. The lisp interpreter knows the following functions: CONS, CAR, CDR, APPEND and binary '+'.

   ```
   >2
   2
   >'(1 2 3)
   (1 2 3)
   >(CAR '(1 2 3))
   1
   >(CDR '(1 2 3))
   (2 3)
   >(+ 1 2)
   3
   >(CONS 1 '(2))
   (1 2)
   ```

   c) add in the language the empty list "()"

   hints:

   - try first to complete the code up to a parser for the given Lisp subset. (without actions for productions). Add actions after that in order to have an interpreter.
   - '(1 2 3) is recognized as $form \Rightarrow l\_form \Rightarrow$ '$(enum)$

3. Analyse a complete interpreter for basic control structures from the folder interpreter. It uses similar structures to if example.