1 Week 7

1. Complete the grammar for *if* statements from previous lab. Use the file if_l6.y included in teh current folder. Add functions from $if\_functions.c$ as actions.

   a) First, compile and analyse the functions defined in $if\_functions.c$.

   ```
   gcc  if_functions.c
   ```

   b) Check the code included in the comments from the first part and uncomment the needed part.

   c) Check the definition of type for the nonterminal symbols.

   d) Once your code is correctly working, eliminate %prec IFX frm if_stmt first rule and check the tree for

   ```
   if(a>2) then if (a >5) then a=1 else a=2
   ```

   Read section 2.6 from Yacc.txt.

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 $form \Rightarrow l\_form \Rightarrow$ '$(enum)$