Your third part of the project is to traverse the tree, checking the types and generating the symbol table. The fields in the symbol table is up to you; I would expect at least the name of the variable, the type, its scope level, and for arrays the upper and the lower limits in the declaration and the basetype (but it is your choice). You must handle scopes, although you may want to start with only one level, make sure it works, and then add the support for multiple scopes. The declarations in a DECLARE section start a scope ending at the corresponding end.

The restrictions in the program are:

- a DECLARE starts a new scope that ends at the corresponding END
- the type of the LHS and the RHS of an assignment must agree;
- the expressions in an IF must have type Boolean;
- for a FOR loop, the 2 values in the range must be the same type and the identifier is (implicitly) declared in a new scope that encases the body of the FOR loop
- the operands for OR, AND, XOR, and NOT must be Boolean, and the result is Boolean;
- the operands for the relational operators must agree, and the result is Boolean;
- the operands for the arithmetic operators (including unary + and -) must be Integer, and the result is Integer;
- the an Ident used with an index ($A(3)$) must be declared as an array variable, and the index must be of type Integer.

Note there 3 places where a scope starts: the start of the program, every DECLARE, and every FOR.

The executable must be called ada, it will take one command-line argument (the input file), and is to output (to stdout) any errors found and just before the end of every scope, the scope level and the names and types of all visible Idents. You must turn in all the source files: the .l, .y files, the files containing the main program and the checking routines, all the needed .h files, and a Makefile with extension .mak.

It is my feeling that you should start with one scope with assignments and expressions; then add the DECLARE construct, WHILE, EXIT, and FOR; and finally the declaration and use of array variables.