Compiler Project : “Lexical and syntax analysis”

Prolog

- A PROLOG program consists of three main section (**Predicates** , **clauses** and **goal**)
- Each section start by its name
- Statement in section clauses must end with dot (.)While statement in section predicates are not.
- Section goal must have only one statement and end with dot(.)
- Section predicates is to define predicates used in clauses and goal sections.
- Predicate is defined as following
    - Predicate_name(data type of parameter1, data type of parameter,…..)
- There is no space in predicate or variable names
- Rules for Name of variable are (must begin with uppercase letter or underscore followed by alphanumeric or underscore)
- Data types are(integer, symbol, char string and real)
- The name of predicates, data types or values of data type *symbol* must begin with lowercase.
- The special variable (anonymous variable) represented by _ and it is always free.
- There is no if statement or loop statement.
- The statement in section clauses are facts or rules.
- Facts is written as predicate name followed by the parameter values
- Rules written as head :- body

- Head is single predicate following by variables or values .
- Body is set of predicates that connected by “and” represented by (,) or by “or” represented by (;)

- Never accept variable in the facts.
- comments
  - for one line used %
  - for multi lines /* */
- There are a readln to read string of characters, readint to read integer and readchar to read single character.
- Write predicate that perform output. It outputs integer variables and string literals, separated by commas String literals are enclosed in double quotes.
- Relational operators are: <,<= >,>= ,= ,<>.
- Arithmetic operators are: +, - , *, and /.
- The PROLOG language has many other features, Any addition to the language specification will be appreciated and you will have a bonus on

Project Requirement

1- Scanner
   a. Design DFA for valid tokens
   b. Implement Scanner using Python.
   c. Visualize DFA for valid tokens via project GUI.
2- Parser
   a. Design Grammar for given language description.
   b. Implement parser.
   c. Visualize the output of Parser as a tree view.

Project Delivery Rules:

1. All the team members should be present during the Project delivery.
2. You’re asked to deliver your Project during your assigned time slot
3. **Delivery Date will be announced.**