Compiler Project : “Lexical and syntax analysis”

LISP Programming Language

- A LISP program are made up of three basic building blocks:
- Atom
    - It is a number or string of contiguous characters. It includes numbers and special characters.
- **Example**

```
hello-from-tutorials-point
name
123008907
*hello*
Block#221
abc123
```

- List
    - A sequence of atoms and/or other lists enclosed in parentheses. Following are examples of some valid lists:
    -

```
( i am a list)
(a ( a b c) d e fgh)
(father tom ( susan bill joe))
(sun mon tue wed thur fri sat)
( )
```

- String is a group of characters enclosed in double quotation marks.
- ***All statements of Lisp written as lists***
- The semicolon symbol (;) is used for indicating a comment line.
    - Example of comment:

```
; this line to display the result
```

- No data type for variables

- the basic arithmetic operations in LISP are +, -, *, /, mod(*for modulus*), rem (*for remainder*) , incf(*for increment*) and decf(*decrement*)
- Relational operators are: <= >= = <>.
- LISP represents a function call f(x) as (f x), for example **cos(45)** is written as **cos 45**
- Relational and basic arithmetic operators written as function
- **Example**
    - (< A B)
    - (* 2 3)

- Expressions are limited to Boolean and arithmetic expressions.
- Boolean expressions are used as tests in control statements
- Parentheses not for grouping but to represent list
- LISP expressions are case-insensitive, cos 45 or COS 45 are same.
- The letter **t**, that stands for logical true. The value **nil**, that stands for logical false, as well as an empty list.
- You can specify the value of variable by function ***setq***

Example: *(setq x 10)*

- There is loop statement **dotimes** : (**dotimes** allows looping for some fixed number of iterations.)
- **Example**

```
(dotimes (n 11)
   (write n) (write (* n n))
```

- **while** In simplest form it is followed by a test clause, and a test action. If the test clause evaluates to true, then the test action is executed otherwise, the consequent clause is evaluated.
- **Example**

```
(when (test-clause) (<action1) )
```

- There are a read and write statements that perform input/output , read any value from user and write string enclosed by double quotation ,variables separated by commas
- Lisp 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.**