**The Tiny Language (Lexical Analyzer)**

**Deadline: 27 Tuesday**

**A program in TINY has a simple structure with the following characteristics (minimum requirements):**

* A sequence of statements separated by semicolons.
* No procedures and no declarations
* All variables are integers which are declared by simply assigning values to them (like BASIC)
* Only two control statements which may include statement sequences:
  + If statement: has an optional else part and must be terminated by the word end.
  + Repeat statement
* Read and Write statements that perform input/output
* Multiline comments ( { }) are used for block comments but comments cannot be nested for simplicity. Supporting nested comments is optional (bonus)
* Expressions are limited to Boolean and arithmetic expressions.
* Arithmetic expressions may involve: integer constants, variables, parentheses and any of the three integer operators -, + and \* with the usual mathematical properties (presedence and associativity)
* Comparison operators are only: < and =
* Boolean expressions only appear as tests in control statements (no Boolean variables, assignment or I/O).

**Important Note:**

***Any addition to the language specification will be appreciated and you will have a bonus on it. For example, you could support variable declarations, add more data types, … etc. All students are encouraged to add new unlisted features and remember that what you hear has a lifetime in your memory which is increased and lasts if you practice it by hand!***

**The Tiny Language Tokens**

|  |  |  |
| --- | --- | --- |
| **Reserved Words** | **Special Symbols** | **Other** |
| If | + | **Number** (1 or more digits) |
| Then | - |
| Else | \* |
| End | = |
| Repeat | < | **Identifier** (1 letter followed by zero or more letters/digits) |
| Until | ( |
| Read | ) | **Comment** { any input except nested comments } |
| Write | ; |
|  | := |

Note that “:=” is the assignement operator while “=” is the equality checking operator.

For convenience, use the following enum as the TokenType of your tokens:

public enum TokenType

{

EndFile, Error,

If, Then, Else, End, Repeat, Until, Read, Write, //Reserved Words

ID, Num, //multicharacher tokens

Assign, Equal, LessThan, Plus, Minus, Times, LeftParentheses, RightParentheses, SemiColon //special symbols

}

Required Output:

* Program read tiny code out :
  + List of tokens
  + Symbol table
  + Comments table
  + Error list
* Program is generic not **ad hoc code**