Raynor Kirkson E. Chavez
PSet 1 Part 1
CS 155 AY 2022-2023
Dr. Susan Pancho-Festin

1. ***T_RES*** *(Keywords)*: procedure, array, node, if, then, elseif, else, case, endcase, while, do, endwhile, repeat, until, loop, forever, for, endfor, input, output, call, return, stop, end, floor, ceil, log, goto
    a. *Regular expression*: `^(procedure|array|node|if|then|elseif|else|case|endcase|while|do|endwhile|repeat|until|loop|forever|for|endfor|input|output|call|return|stop|end|floor|ceil|log|goto|by)$`

2. ***T_UnaryOp*** *(Unary Operators)*: not
    a. Regular expression: `^(not)$`

3. ***T_ExpoOp*** *(Exponentiation)*: ^
    a. Regular expression: `^(\^)$`

4. ***T_MDOp*** *(Multiplication and Division)*: * (we convert the multiplication symbol x to *, to avoid confusion with the variable name x), /, mod
    a. Regular expression: `^(\*|\/|mod)$`

5. ***T_ASOp*** *(Addition and Subtraction)*: +, -
    a. Regular expression: `^(\+|-)$`

6. ***T_RelOp*** *(Relational Operators)*: ==, <=, >=, !=,<, >
    a. Regular expression: `^(==|<=|>=|!=|<|>)$`

7. ***T_LogicOp*** *(Logical Operators)*: and, or
    a. Regular expression: `^(and|or)$`

8. ***T_AssignOp*** *(Assignment)*: =
    a. Regular expression: `^(=)$`

9. ***T_Delim*** *(Delimiters)*:   ;, :, (,), ', [ ], { }, ,
    a. *Regular expression*: `^([,:;\(\)\[\]\{\}])$`

10. ***T_ID*** *(Identifiers)*: variable and function names, e.g., middle, lower, upper, BINARY_SEARCH, A, n, x
    a. *Regular expression*: `^([a-zA-Z_][a-zA-Z0-9_]*)$`
11. ***T_NumLit*** *(Numeric literals):* integers, floating point numbers
    a. Regular expression: `^(-?([0-9]+(\.[0-9]+)?))$`

12. ***T_StrLitD*** *(String literals):*
    a. Regular expression: `\"([^\"]|\\.)*?\"`   (for string in double quotes)
13. ***T_StrLitS*** *(String literals):*
    a. Regular expression: `\'([^\']|\\.)*?\'`   (for string in single quotes)

Raynor Kirkson E. Chavez
PSet 1 Part 1
CS 155 AY 2022-2023
Dr. Susan Pancho-Festin

Token Class	**Symbols or Expected Values**	**Regex**
T_RES *(Keywords)*	procedure, array, node, if, else, case, endcase, while, do, endwhile, repeat, until, loop, forever, for, endfor, input, output, call, return, stop, end, floor, ceil, log, goto, by	*^(procedure|array|node|if|then|elseif |else|case|endcase|while|do|endwhile |repeat|until|loop|forever|for|endfor |input|output|call|return|stop|end |floor|ceil|log|goto|by)$*
T_UnaryOp *(Unary Operators)*	not	*^(not)$*
T_ExpoOp *(Exponentiation)*	^	*^(\^)$*
T_MDOp *(Multiplication and Division)*	x, /, mod	*^(*|\/|mod)$*
T_ASOp *(Addition and Subtraction)*	+, -	*^(\+|-)$*
T_RelOp *(Relational Operators)*	*==, <, >, <=, >=, !=*	*^(==|<=|>=|!=|<|>)$*
T_LogicOp *(Logical Operators)*	*and, or*	*^(and|or)$*
T_AssignOp *(Assignment)*	=	*^(=)$*
T_Delim *(Delimiters)*	*;, :, (,), ', [ ], { }*	*^([,:;\(\)\[\]\{\}])$*
T_ID *(Identifiers)*	variable and function names	*^([a-zA-Z_][a-zA-Z0-9_]*)$*
T_NumLit *(Numeric literals)*	integers, floating point numbers	*^(-?([0-9]+(\.[0-9]+)?))$*
T_StrLitD *(String literals in “”)*	*strings in “”*	*\"([^\"]|\\.)*?\"*
T_StrLitS *(String literals in ‘’)*	*strings in ‘’*	*\'([^\']|\\.)*?\'*

Table 1. Summarized Token Classes in order of precedence