**PROYECTO #1. ANALIZADOR LÉXICO PARA EL LENGUAJE GRM123**

|  |
| --- |
| **Desarrollo: Individual**, en el Lenguaje de Programación que usted elija.  **Entrega:** Enviar la carpeta del proyecto a [ginobarroso@uagrm.edu.bo](mailto:ginobarroso@uagrm.edu.bo) hasta el Dom 09/julio/2023, 23:59:59 hrs. |

|  |
| --- |
| **Defensa:** Mie 19/julio/2023, 11:30 hrs, en el Lab 236-45  **Importante:** No se olvide de llevar el dibujo de su Diagrama de Transiciones (dt). |

Desarrollar un Analizador Léxico (Analex) para el lenguaje GRM123, el cual es un híbrido entre C y Pascal, tomando en cuenta que el mismo es **case-sensitive**. Los tokens a reconocer son:

|  |  |
| --- | --- |
| **LEXEMA(S)** | **TOKEN** |
| "" (Cadena vacía) | < FIN,\_> |
| (El lexema que dio el error) | < ERROR,\_> |
| “main” | < MAIN,\_> |
| “void” | < VOID,\_> |
| “if” | <IF,\_> |
| “else” | <ELSE,\_> |
| “for” | <FOR,\_> |
| “to” | <TO,\_> |
| “while” | <WHILE,\_> |
| "repeat" | <REPEAT,\_> |
| "until" | <UNTIL, \_> |
| “readln” | <READLN,\_> |
| “print” | <PRINT,\_> |
| “println” | <PRINTLN,\_> |
| “,” | <COMA,\_> |
| “;” | <PTOCOMA,\_> |
| ":" | <DOSPUNTOS,\_> |
| “(“ | <PA,\_> |
| “)” | <PC,\_> |
| “{“ | <LA,\_> |
| “}” | <LC,\_> |
| “:=” | <ASSIGN,\_> |
| “not” y “!” | <NOT,\_> |
| “and” y “&” | <AND,\_> |
| “or” y “|” | <OR,\_> |
| “false” | <FALSE,\_> |
| “true” | <TRUE,\_> |
| “+” | <MAS,\_> |
| “-“ | <MENOS,\_> |
| “\*” | <POR,\_> |
| “mod” y “%” | <MOD,\_> |
| “div” y “/” | <DIV,\_> |

|  |  |
| --- | --- |
| **LEXEMA(S)** | **TOKEN** |
| "++" | <INC, \_> |
| "--" | <DEC, \_> |
| “return” | <RETURN, \_> |
| (números enteros sin signo) | <NUM, valor> |
| (Empiezan con Letra y luego le sigue una combinación de Letras y Dígitos) | **<ID, -1>** |
| (chars encerrados entre comillas “..”, sin EOLN ni EOF) | **<STRINGctte, 0>** |
| “=”  “<”  “>”  “<=”  “>=”  “!=” | <OPREL,IGUAL>  <OPREL,MEN>  <OPREL,MAY>  <OPREL,MEI>  <OPREL,MAI>  <OPREL,DIS> |
| "string"  “boolean”  “int” | <TIPO, STRING>  <TIPO, BOOLEAN>  <TIPO, INT> |

Puesto que su Analex no trabajará con la Tabla de Símbolos, para el token **ID** use como atributo –1 , y para el **STRINGctte,**  el 0: **<ID, -1>, <STRINGctte, 0>.**

|  |
| --- |
| **Comentarios**. Los comentarios que su Analex, debe tratar son:   * Comentarios de línea: //… * Comentarios multilínea: /\* … \*/ |

*Ejemplo de un programa, escrito en el lenguaje GRM123. Las palabras reservadas (Keyword’s), se escriben en color azul:*

**var** sw **: boolean;** //Solo trabaja con variables globales.

**var** altura, N : **int**; //Se permite una sola línea de declaración, después de **var**.

**var** s, str, name : **string**; //Solo se usan los tipos boolean, int y string.

**void** lectura(){

sw := **true**;

**repeat** /\* El repeat-until, no usa llaves \*/

**print**("Introduzca N: "); //Se usa **print** y **println** para salida en consola.

**readln**(N);

sw := (N >= 0);

**if** (!sw)

**println**("El valor ", N, " es incorrecto");

**else**

**return**;

**until** (**false**);

}

**var** i, f : **int**;

**void** factorial(){

f := 1;

**for** i:=1 **to** N{

f := f \* i;

}

}

**void** factorial2(){

i := 2;

i--;

f := 1;

**while** (i<=N){

f := f \* i;

i++;

}

}

**main**(){ //**main** es una palabra reservada (no se antepone la palabra **void**).

lectura();

factorial();

**println**("El factorial de ", N, " es ", f);

}