

Práctica 1

Analizador Léxico

Dado un programa en lenguaje PsiCoder, su tarea consiste en realizar el análisis léxico. Por cada componente léxico se debe imprimir una línea con toda la información del token identificado. A continuación se muestra la manera correcta de generar las salidas correspondientes.

- Las palabras reservadas se deben imprimir en el siguiente formato: **<palabra_reservada,línea,columna>**.

Ejemplo:

```
funcion_principal  
    entero  
    fin_principal
```

debe imprimir las siguientes tres líneas (una por cada token):

```
<funcion_principal,1,1>  
<entero,2,3>  
<fin_principal,3,5>
```

Observe que en este punto el programa podría no ser válido a nivel sintáctico, semántico o lógico. Sin embargo, la tarea actual se centrará únicamente en el análisis léxico.

Los comandos especiales **leer** y **escribir** deben tratarse como palabras reservadas del lenguaje.

- Los identificadores se deben imprimir en el siguiente formato: **<id,lexema,fil,a,columna>**

Ejemplo:

```
funcion_principal  
    entero a  
    booleano z  
    fin_principal
```

debe imprimir:

```
<funcion_principal,1,1>  
<entero,2,5>
```

<id,a,2,12>

<booleano,3,5>

<id,z,3,14>

<fin_principal,4,1>

- De forma similar al caso anterior, se deben imprimir las cadenas no específicas correspondientes a números enteros, reales, cadenas de caracteres y caracteres. El formato será el que sigue a continuación, respectivamente:

<tk_entero,lexema,fila,columna>

<tk_real,lexema,fila,columna>

<tk_cadena,lexema,fila,columna>

<tk_caracter,lexema,fila,columna>

El lexema de las cadenas de caracteres debe estar entre comillas dobles, por ejemplo:

<tk_cadena,"Lenguajes de Programacion",1,10>

El lexema de los caracteres debe estar entre comillas sencillas, por ejemplo:

<tk_caracter,'Z',2,5>

Nótese que para el caso de valores booleanos, como sólo pueden tener el valor de 'falso' o 'verdadero', estas palabras deben ser tratadas como palabras reservadas.

- Los operadores y símbolos especiales se deben imprimir en el siguiente formato:

<token,fila,columna>

Donde el **token** será determinado de acuerdo a la siguiente tabla:

Símbolo	Token
+	tk_mas
-	tk_menos
*	tk_mult
/	tk_div
%	tk_mod
=	tk_asig

<	tk_menor
>	tk_mayor
<=	tk_menor_igual
>=	tk_mayor_igual
==	tk_igual
&&	tk_y
	tk_o
!=	tk_dif
!	tk_neg
:	tk_dosp
'	tk_comilla_sen
"	tk_comilla_dob
;	tk_pyc
,	tk_coma
(tk_par_izq
)	tk_par_der

- Los comentarios (de una sola línea o de múltiples líneas) no generan ninguna salida, serán ignorados.
- **Errores:** cuando el analizador detecte un error léxico debe abortar el análisis y debe reportar el error a la salida mediante un mensaje con el siguiente formato:

>>> Error lexico (linea: X, posicion: Y)

Donde X y Y son números enteros que representan el número de línea y la posición donde fue detectado el inicio del error léxico.

Entrada

Para probar el analizador léxico se evaluarán distintos casos de prueba. Cada caso de prueba será pasado al programa por la entrada estándar. Cada entrada consiste en un programa escrito en el lenguaje PsiCoder.

Salida

Por cada archivo de entrada se debe mostrar la salida según lo especificado anteriormente.

Ejemplos

Entrada	Salida
<pre>funcion_principal imprimir (3+5); fin_principal</pre>	<pre><funcion_principal,1,1> <imprimir,3,5> <tk_par_izq,3,14> <tk_entero,3,3,15> <tk_mas,3,16> <tk_entero,5,3,17> <tk_par_der,3,18> <tk_pyc,3,19> <fin_principal,5,1></pre>

Entrada	Salida
<pre>funcion_principal entero a1 = 5 * 3; imprimir(a1 / 10); fin_principal // comentario al final</pre>	<pre><funcion_principal,1,1> <entero,2,5> <id,a1,2,12> <tk_asig,2,15> <tk_entero,5,2,17> <tk_mult,2,19> <tk_entero,3,2,21> <tk_pyc,2,22> <imprimir,3,5> <tk_par_izq,3,13> <id,a1,3,15> <tk_div,3,18> <tk_entero,10,3,20> <tk_par_der,3,23> <tk_pyc,3,24> <fin_principal,4,1></pre>

Entrada	Salida
<pre> /* esto no debe importar, pero cuenta las lineas */ funcion entero sum(entero a, entero b) hacer retornar a + b; // fin fin_funcion </pre>	<pre> <funcion,4,1> <entero,4,9> <id,sum,4,16> <tk_par_izq,4,19> <entero,4,20> <id,a,4,27> <tk_coma,4,28> <entero,4,30> <id,b,4,37> <tk_par_der,4,38> <hacer,4,40> <retornar,5,5> <id,a,5,14> <tk_mas,5,16> <id,b,5,18> <tk_pyc,5,19> <fin_funcion,7,1> </pre>

Entrada	Salida
<pre> /* esto no debe importar, pero cuenta las lineas */ funcion entero sum(entero a, entero b) hacer retornar a + Ñ; // fin fin_funcion </pre>	<pre> <funcion,4,1> <entero,4,9> <id,sum,4,16> <tk_par_izq,4,19> <entero,4,20> <id,a,4,27> <tk_coma,4,28> <entero,4,30> <id,b,4,37> <tk_par_der,4,38> <hacer,4,40> <retornar,5,5> <id,a,5,14> <tk_mas,5,16> >>> Error lexico (linea: 5, posicion: 18) </pre>