

**Objetivo**

Elaborar un analizador léxico en *lex/flex* que reconozca los componentes léxicos pertenecientes a las clases abajo descritas.

**Descripción**

Las clases de los componentes léxicos válidos para el analizador léxico son:

Clase	Descripción
0	Palabras reservadas (ver tabla).
1	Identificadores. Iniciar con \$ y le sigue al menos una letra minúscula o mayúscula. Ejemplos: \$ejemplo, \$Variable, \$OtraVariable, \$XYZ
2	Constantes numéricas enteras. En base 10 (secuencia de dígitos del 0-9 sin 0's a la izquierda, excepto para el número 0), en base 8 (inicien con O u o y le sigan dígitos del 0 al 7).
3	Constantes numéricas reales. Siempre deben llevar parte decimal y es opcional la parte entera. Ejemplos: 73.0, .0, 10.2 No aceptados: . , 12 , 4.
4	Constantes cadenas. Encerrado entre comillas (") cualquier secuencia de más de un carácter que no contenga " ni '. Para cadenas de un solo carácter, encerrarlo entre apóstrofes ('). La cadena de unas comillas debe ser encerrada entre apóstrofes: ""'. La cadena de un apóstrofo debe ser encerrada por comillas: ""'. No se aceptan cadenas vacías. Ejemplos NO válidos: "ejemplo no "valido" , "" , "" , "hola 'mundo"
5	Símbolos especiales [ ] ( ) { } , : ;
6	Operadores aritméticos + - * / % \ ^
7	Operadores relacionales (ver tabla).
8	Operador de asignación =

El número de clase es inamovible.

- El analizador léxico tendrá como entrada un archivo con el programa fuente, **el cual se indicará desde la línea de comandos** al momento de mandar a ejecutar el analizador léxico.
- Como delimitador de un componente léxico será uno o varios espacios, tabuladores o saltos de línea, **así como el inicio de otro componente léxico**. Considerar el orden en las acciones en el programa lex/flex

- Los tokens se representarán en una estructura con dos campos:  
campo1: la clase  
campo2: el valor (de acuerdo con las siguientes tablas e indicaciones)

Valor	Palabra reservada	Equivalencia en C
0	alternative	case
1	big	long
2	evaluate	if
3	instead	else
4	large	double
5	loop	while
6	make	do
7	number	int
8	other	default
9	real	float
10	repeat	for
11	select	switch
12	small	short
13	step	continue
14	stop	break
15	symbol	char
16	throw	return

Valor	Op. relacional
0	<
1	>
2	<=
3	>=
4	==
5	!=

- El valor para el token de cada identificador es la posición dentro de la tabla de símbolos. Para los operadores relacionales y las palabras reservadas, el valor del token será la posición en su correspondiente tabla (catálogo).
- Las constantes numéricas reales se incluirán en una tabla de literales, las cadenas tendrán su propia tabla de literales; por lo que el valor de su token será la posición dentro de la tabla de literales correspondiente. Cada que el Analizador Léxico encuentre una cadena o constante numérica real, revisará si ya se encuentra en su tabla de literales si existe no la inserta, indicar método de búsqueda. Para la clase 5 el valor en su token es el mismo carácter o su correspondiente ASCII.
- Cuando detecte un error léxico, deberá seguir el reconocimiento a partir del siguiente símbolo.
- El analizador deberá crear la Tabla de Símbolos para almacenar a los identificadores. Esta tabla manejará los campos: posición, nombre del identificador y tipo (este último será de tipo entero y podrán ponerle como valor inicial -1). Se indicará en el documento a entregar, el tipo de estructura de datos empleada, así como el método de búsqueda a utilizar.
- Las Tablas de Literales, una para las constantes numéricas reales y otra para las cadenas, deberán tener como estructura, dos campos: la posición y el dato (cadena o constante numérica real) según el caso.
- Al término del análisis léxico deberá mostrar la tabla de símbolos, las tablas de literales, así como los tokens. También podrán almacenarse en archivos para su mejor revisión.

- Los errores que vaya encontrando el analizador léxico, los podrá ir mostrando en pantalla o escribirlos en un archivo. Deberá recuperarse de los errores encontrados para continuar con el reconocimiento de todos los componentes léxicos del archivo de entrada.
- El programa deberá estar comentado, con una descripción breve de lo que hace (puede ser el objetivo indicado en este documento), el nombre de quién(es) elaboró(aron) el programa y fecha de elaboración, así como lo que hace cada función. Se deberá cuidar mucho la sangría que denota la dependencia de instrucciones.

### Entregar:

Un documento con la siguiente estructura:

1. Descripción del problema (no del programa), incluyendo características que deben cumplir los componentes léxicos que reconocerá el analizador léxico y **la expresión regular de cada clase**. Forme su diagrama y tabla de transiciones.
2. Propuesta de solución y fases del desarrollo del sistema: Análisis (planificación, indicando participantes por cada actividad), Diseño e Implementación. En el diseño indicar exactamente cómo se definirá la tabla de símbolos, las tablas de literales y los tokens, la técnica de búsqueda e inserción de los identificadores y la técnica de inserción de las cadenas y constantes numéricas reales.
3. Indicaciones de cómo correr el programa.
4. Conclusiones por cada participante.

**Nota: se podrá elaborar individualmente o en equipo de 2 a 3 personas.**

Colocar el documento y sólo el programa fuente definitivo a la plataforma educativa (apartado tareas y empaquetados en zip) y en su caso, sólo un miembro del equipo.

**Fecha de entrega: 13 marzo 2024.**