

Primer programa
Analizador Léxico**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	Operadores aritméticos + - * / \$
2	Operadores de asignación (ver tabla)
3	Símbolos especiales () { } [] & , :
4	Operadores relacionales (ver tabla)
5	Identificadores. Inicien con letra (mayúscula o minúscula), le sigan letras o dígitos (hasta cinco) y terminen con guion bajo.
6	Constantes numéricas enteras (base 10, sin sufijos) signados o no signados, máximo 6 dígitos, mínimo 1. Para no confundir el signo con un operador aritmético se dejará o no un espacio en los siguientes casos: -56+ +46 56- 41 72+a_
7	Constantes numéricas reales. No signados. Casos: 2.67 .85 31. Al definir la expresión regular, no debe incluir la cadena. como constante real.
8	Constantes cadenas. Iniciar y terminar con comillas dobles, cualquier carácter excepto las comillas dobles. Tamaño entre 2 y 40 (considerando las comillas).
9	Constante carácter. Cualquier carácter encerrado entre comillas sencillas (apóstrofes).

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 caso de las constantes numéricas enteras.

- 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
0	cadena
1	caracter
2	else
3	entero
4	for
5	if
6	real
7	return
8	void
9	while

Valor	Op. relacional	significado
0	$\wedge\wedge$	mayor que
1	\wedge''	menor que
2	$==$	igual que
3	$\wedge=$	mayor o igual
4	$\wedge''=$	menor o igual
5	$<>$	diferente

Valor	Op. asignación	significado
0	\sim	igual
1	$+ \sim$	más igual
2	$- \sim$	menos igual
3	$* \sim$	por igual
4	$/ \sim$	entre igual
5	$\$ \sim$	módulo igual

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 enteras y 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, NO revisará si ya se encuentra en su tabla de literales, simplemente la insertará.

Para las clases 1 y 3, 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 y otra para las cadenas, deberán tener como estructura, dos campos: la posición y el dato (cadena o constante numérica) 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:

- 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**.
- 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.
- Indicaciones de cómo correr el programa.
- Conclusiones por cada participante.

Nota: se podrá elaborar individualmente o en equipo de 2

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

Fecha de entrega: 19 de octubre de 2021.

Ejemplo de reconocimiento.

Contenido del archivo a analizar:

```
cadena a_,b2,c1_  
if x_ ^^ (-23+c1_)
```

Salidas que debe arrojar el analizador:

Tabla de Símbolos Tabla de literales(numéricas)

0 a_ 0 -23

1 c1_

2 x_

Tokens

(0,0)

(5,0)

(3,,)

(3,,)

(5,1)

(0,4)

(5,2)

(4,0)

(3,()

(6,0)

(1,+)

(5,1)

(3,))

Errores:

b2 no reconocido

; no reconocido.