Universidad de Málaga
Departamento de Lenguajes y
Ciencias de la Computación
Campus de Teatinos, 29071 MÁLAGA

TRADUCTORES, COMPILADORES E INTÉRPRETES. 09/10
EJERCICIOS DE LEX. TEMA 2.

1.) Hacer un programa LEX que convierta todos los caracteres de su entrada que sean LF (ascii 10), por LF y CR (ascii 10 y ascii 13).

2.) Hacer un programa LEX que tras leer un texto nos diga el número de caracteres, palabras y líneas de dicho texto, entendiéndose por palabra toda secuencia de caracteres que no posea ni espacios ni tabuladores ni retornos de carro (por tanto las comas, arrobas y demás se consideran parte de las palabras).

3.) Hacer un programa en LEX, de manera que se cifre el texto de entrada. La salida debe ser exactamente igual a la entrada (recuérdese que cuando un lexema no entra por ningún patrón se emite tal cual por pantalla), pero convirtiendo cada palabra en su inversa. El concepto de palabra es el mismo que en el ej. 2).

4.) Hacer un programa LEX que reconozca números naturales múltiplos de dos. No utilizar la función **módulo** de C. Todos los demás lexemas deben ignorarse. Al final, el programa nos debe decir cuántos múltiplos de dos se han detectado.

5.) Exactamente igual al anterior, pero que reconozca múltiplos de cuatro.

6.) Hacer un programa LEX que lea un fichero escrito en Java e indique lo siguiente:
* Cuántos comentarios hay (sólo multilínea).
* Cuántas veces aparece la palabra **public**.
* Cuántas veces aparece la palabra **private**
* Cuántos espacios y tabuladores hay.

7.) Hacer un programa LEX que lea un fichero escrito en Java e indique la profundidad de anidamiento de llaves más grande. Cada vez que unas llaves están contenidas dentro de otras, aumenta el nivel de anidamiento.

8.) Hacer un programa Lex que reconozca los siguientes tipos de lexemas:
- Números naturales precedidos (opcionalmente) de un signo más o menos.
- Números con coma decimal. Cada alumno puede asumir el formato que desee: con coma o punto decimal, con o sin exponente, etc. En cualquier caso, delante y detrás del punto/coma decimal debe haber al menos un dígito.
- Cadenas entrecomilladas.
- Caracteres entre comillas simples.
- Las palabras reservadas: DECLARATION, BEGIN y END.
- Identificadores de usuario en el normal sentido de la palabra.
- Espacios, tabuladores y retornos de carro.
- Cualquier otro carácter.

Al final, el programa debe indicar por pantalla cuantos lexemas de cada tipo ha reconocido.

9.) Mediante un programa LEX, sacar un listado con todos los identificadores de usuario de un fichero escrito en Java. Para ello, hay que reconocer todas las palabras reservadas, y todo identificador que no sea una palabra reservada se asume que es un identificador de usuario. Los identificadores se sacan sobre la marcha, de forma que un mismo identificador puede salir varias veces. La tarea práctica nº 3 tiene cierta similitud con este ejercicio.

10.) Exactamente igual al anterior, pero los identificadores no se deben sacar por pantalla sobre la marcha, sino que deben almacenarse en una lista y, al final del funcionamiento de **yylex()**, emitir por pantalla el contenido de dicha lista.

11.) Supuesto que se tiene un diccionario de palabras en formato texto, (almacenado en un fichero con una palabra por línea), procesar mediante un programa LEX, cualquier texto de entrada, visualizando por pantalla todas las palabras que no estén en dicho diccionario. El diccionario puede ser volcado a memoria justo antes de comenzar el procesamiento, esto es, antes de llamar a **yylex()**.

12.) Realizar un programa con PCLex que tome por entrada un fichero de texto en formato .srt (subtítulos de películas: http://en.wikipedia.org/wiki/SubRip) y produzca lo mismo como salida pero aumentando cada bloque **hours:minutes:seconds,milliseconds** en 2 segundos y 870 milisegundos (2,870 segs.). Por ejemplo, el bloque:

```
1
00:00:20,000 --> 00:00:24,400
Los crimenes cometidos por
el imperialismo yanqui en la guerra del 98,

2
00:00:24,600 --> 00:00:27,800
deben resarcirse ...

3
00:00:59,600 --> 00:01:59,800
La humillación sufrida por aquella debacle ...
```

Debe emitirse como:

```
1
00:00:22,870 --> 00:00:27,270
Los crimenes cometidos por
el imperialismo yanqui en la guerra del 98,

2
00:00:27,470 --> 00:00:30,140
deben resarcirse ...

3
00:01:02,470 --> 00:02:02,670
La humillación sufrida por aquella debacle ...
```

Nótese el acarreo de milisegs. a segs., de segs. a minutos, etc.