



PRUEBA T1

Traductores, compiladores e intérpretes

Curso 08/09

Nombre y apellidos: _____
Grupo (mañana - tarde). Táchese lo que no proceda.

Se desea construir un programa Lex que reconozca los siguientes patrones y realice las acciones indicadas:

1. Espacios, tabuladores o retornos de carro que deben ignorarse.
2. Comentarios que comienzan por una doble barra de división y se supone que acaban al finalizar la línea. Deben ignorarse
3. Las palabras reservadas IF, WHILE, OR, XOR, NAND y FOR. Deben ignorarse
4. Identificadores de usuario que sólo pueden estar formados por letras mayúsculas o minúsculas y por el símbolo de subrayado. Debe contarse cuántos identificadores aparecen en el texto de entrada; un mismo identificador se contará tantas veces como aparezca. Al final del análisis debe aparecer el total de identificadores encontrados.
5. Números decimales. Un número decimal debe tener al menos un dígito a cada lado del punto decimal. Deben sumarse los valores de estos números y emitir la suma total al final del análisis léxico.
6. Cualquier otra cosa que se reconozca debe emitirse por pantalla avisando de que se trata de un error. Además hay que indicar el número de línea en que se encuentra el fallo.

Al finalizar la función yylex() debe indicarse por pantalla el número de identificadores que se han encontrado y la suma total de valores decimales.

Un ejemplo de entrada y salida sería:

Entrada	Salida
// Un ejemplo // AQUI SE EMPIEZA FOR AND , 23.7 43.99 FOR K 3 OR FOR ALFA BETA ALFA EFORD 23.8 A , WHILE // FOR	Fallo linea: 3 con , Fallo linea: 5 con 3 Fallo linea: 15 con , El numero de IDs es: 6 La suma de valores es: 91.49



PRUEBA T2
Traductores, compiladores e intérpretes
Curso 08/09

Nombre y apellidos: _____
Grupo (mañana - tarde). Táchese lo que no proceda.

Se desea construir los programas Lex y YACC destinados a reconocer listas de números naturales separados por comas que constituyan secuencias de Fibonacci. Una secuencia de Fibonacci debe estar formada por, al menos, dos números de manera que cada número (del tercero en adelante debe ser igual a la suma de los dos anteriores. Por ejemplo:

Secuencia	¿Es de Fibonacci?
2, 3, 5, 8, 13, 21	Sí
1, 1, 2, 3, 7, 10, 17	No
10, 1, 11, 12, 23, 35, 58	Sí
5	Error de sintaxis

El programa, tras leer la secuencia, debe emitir por pantalla sencillamente la palabra SÍ o NO indicando, respectivamente, si la secuencia leída es o no de Fibonacci.



Nombre y apellidos: _____

Grupo (mañana - tarde). Táchese lo que no proceda.

Como fase previa a la construcción de una compleja calculadora, se desea construir un intérprete que reconozca expresiones aritméticas en las que intervienen tan sólo los operadores +, * y **menos unario**, así como los **paréntesis**. Asimismo, sólo se permite el uso de **constantes numéricas**.

Para comprender mejor el funcionamiento del analizador, se desea obtener por pantalla, no sólo el resultado final de la operación, sino además, todos y cada uno de los resultados parciales que se van obteniendo. P.ej., ante la entrada

$$3+4*5+(6+2*5)$$

se debe obtener como salida:

$$4*5 = 20$$

$$3+4*5 = 23$$

$$2*5 = 10$$

$$6+2*5 = 16$$

$$3+4*5+(6+2*5) = 39$$

Resultado: 39

Con este objetivo en mente, se pide construir el programa LEX que implementa el analizador lexicográfico, y completar el programa YACC que se da a continuación.

Nota: Se recomienda el uso de la función **sprintf**.

Fichero EJEM9L.LEX

Fichero EJEM9Y.YAC

```
% {  
#include <stdio.h>  
#include <stdlib.h>  
#include <string.h>  
  
typedef struct {  
    int valor;  
    char * texto;  
} expresion;  
%}  
  
%union {  
    int numero;  
    expresion comun;  
}
```

%%