

Programa 1: Análisis léxico

Código fuente: *matrícula_c1.c / matrícula_c1.cpp*

El programa consiste en la implementación de la etapa del análisis léxico de un compilador para el lenguaje de programación visto en clase.

La lista de tokens es la siguiente:

Nombre del token	Especificación léxica	Notas adicionales
INCLUDE	include	
FUNCTION	function	
VAR	var	
IF	if	
ELSE	else	
RETURN	return	
PRINT	print	
MAS	+	
MENOS	-	
POR	*	
ENTRE	/	
ASIGNACION	=	
MENOR_QUE	<	
MENOR_IGUAL_QUE	<=	
MAYOR_QUE	>	
MAYOR_IGUAL_QUE	>=	
IGUAL_QUE	==	
DIFERENTE_QUE	!=	
PARENTESIS_IZQ	(
PARENTESIS_DER)	
LLAVE_IZQ	{	
LLAVE_DER	}	
COMA	,	
PUNTO_COMA	;	
LITERAL_ENTERA	<i>Secuencia contigua de dígitos</i>	La literal termina al encontrar el primer caracter que no sea dígito, sin incluirlo
LITERAL_CADENA	<i>Secuencia de caracteres delimitada entre "</i>	No se reconocen literales de cadena que no estén cerradas; las comillas

		forman parte de la literal
IDENTIFICADOR	<i>Comienza con letra o guión bajo y continúa con cero o más letras, dígitos o guiones bajos</i>	
DESCONOCIDO	<i>Símbolos no reconocidos diferentes de caracteres espacio</i>	El análisis de la entrada termina prematuramente con el primer token de este tipo
FIN_ARCHIVO	<i>El valor EOF de la plataforma almacenado en un tipo char</i>	Marcador almacenado implícitamente como último carácter de la entrada
	<i>Secuencia de caracteres delimitados entre /* y */</i>	Se reconocen comentarios no cerrados y el procesamiento de la entrada continúa en el fin de archivo; los delimitadores forman parte del comentario

Los tokens se reconocen siguiendo el principio de la coincidencia más larga, es decir <= se reconoce como MENOR_IGUAL QUE y no como MENOR_QUE seguido de ASIGNACION.

Entrada: La entrada consiste en código fuente que deberá ser leído desde un archivo llamado codigo.txt.

Salida: Para cada token se debe imprimir lo indicado en la siguiente tabla seguido de un salto de línea. Para LITERAL_ENTERA, LITERAL_CADENA e IDENTIFICADOR se debe imprimir adicionalmente la subcadena reconocida como se indica en la tabla.

Nombre del token	Salida correspondiente
INCLUDE	0
FUNCTION	1
VAR	2
IF	3

ELSE	4
RETURN	5
PRINT	6
MAS	7
MENOS	8
POR	9
ENTRE	10
ASIGNACION	11
MENOR_QUE	12
MENOR_IGUAL_QUE	13
MAYOR_QUE	14
MAYOR_IGUAL_QUE	15
IGUAL_QUE	16
DIFERENTE_QUE	17
PARENTESIS_IZQ	18
PARENTESIS_DER	19
LLAVE_IZQ	20
LLAVE_DER	21
COMA	22
PUNTO_COMA	23
LITERAL_ENTERA	24 <i>literal</i>
LITERAL_CADENA	25 <i>literal</i>
IDENTIFICADOR	26 <i>identificador</i>
DESCONOCIDO	27
FIN_ARCHIVO	28

Ejemplo de entrada	Ejemplo de salida
if if_ 453452 /*hola*/	3
elseeee == = 22 else 333 _	26 if_
include	24 453452
	26 elseeee
	16
	11
	24 22
	4
	24 333
	26 _
	0
	28

Su programa sólo debe imprimir lo solicitado. El código fuente deberá ser enviado como archivo adjunto al correo al203305906@alumnos.azc.uam.mx. No se recibirán ejecutables y de ninguna otra forma.