



Una empresa de software que está desarrollando un lenguaje de programación nos solicitó un programa en lenguaje C que permita realizar un análisis sintáctico preliminar de las líneas de código correspondientes a las declaraciones de variables simples de dicho lenguaje.

Ejemplos de declaración de variables simples: `char car;`
`int i, j, k;`

El programa deberá permitir el ingreso por teclado del texto que incluye una línea de código con la declaración de variables simples por medio de la función **carga**, luego el programa deberá separar en palabras dicha línea de código utilizando la función **separa**, y por último tiene que invocar a la función **analiza** para determinar si es correcta o no la sintaxis de la línea de código referida a la declaración de variables simples, mostrando desde el main la línea de código ingresada y a su lado el resultado del análisis (Sintaxis Correcta ó Sintaxis Incorrecta).

Los prototipos de las funciones son los siguientes:

➤ **char * carga (void); PROGRAMAR**

permite ingresar por teclado la línea de código a analizar.

Devuelve un puntero al comienzo del texto ingresado (máximo 128).

➤ **void separa (char *, char [40][10], int *); NO PROGRAMAR**
recibe como parámetros la dirección de comienzo de la línea de código. Separa en palabras la línea y deja disponible en el main un arreglo bidimensional con la lista de palabras y la cantidad de palabras separadas. Por ejemplo, dada la línea `int nro1, nro2, nro3;` la función deja disponible (además de la cantidad de palabras) la matriz que tiene el siguiente formato:

	0	1	2	3	4	5	6	7	8	9
0	i	n	t							
1	n	r	o	1						
2	,									
3	n	r	o	2						
4	,									
5	n	r	o	3						
6	;									

➤ **char * analiza (char [][], int); PROGRAMAR**

recibe la dirección de comienzo del arreglo bidimensional y la cantidad de palabras separadas; analiza y devuelve el mensaje "Sintaxis Correcta" ó "Sintaxis Incorrecta".

La línea de código será correcta si cumple los siguientes requisitos:

- La fila 0 de la matriz debe contener una palabra correspondiente a un tipo de datos válido. Para ello invocar a la función **int tipo (char *); NO PROGRAMAR** que recibiendo el tipo de datos devuelve 1 si es un tipo de datos válido y 0 si no lo es.
- La última fila de la matriz debe contener un ; (punto y coma).
Para las filas intermedias de la matriz, es decir, exceptuando las filas inicial y final, si existiesen, se debe cumplir que:
- Las posiciones impares deben tener un nombre de identificador válido (entre 1 y 8 caracteres alfanuméricos).
Para ello invocar a la función **int identifica (char *); PROGRAMAR** que devuelve 1 si es un identificador válido y 0 si no lo es.
- Las posiciones pares deben contener una , (coma).

NOTA: Como se desprende del enunciado la declaratoria es para variables de un mismo tipo de dato y no ocupará más de 10 filas de la matriz generada.

PROGRAMAR LA FUNCION main Y LAS INDICADAS (carga, analiza e identifica).
ESCRIBIR DE UN SOLO LADO DE LA HOJA.