

PRÁCTICA INDIVIDUAL DE LEX

Alumno: Daniel Bolaños Martínez

Asignatura: Modelos de Computación (Prácticas)

- **Presentación del problema:**

El problema inicial se basa en un programa de reconocimiento de boletos premiados de un sorteo cualquiera similar al de Navidad. El programa, a partir del analizador léxico de Lex, debe reconocer tres conjuntos de datos:

1. La lista de boletos que han sido premiados en el sorteo.
2. La lista de administraciones de lotería que han vendido los boletos, para simplificar el problema solo trataremos los puestos oficiales.
3. El conjunto de participantes que veamos conveniente.

Cada conjunto estará compuesto de unos datos y restricciones, que se almacenarán internamente en memoria con ayuda de las estructuras de datos pertinentes. Se creará a partir de esta información, una base de datos, sobre la cual se podrá hacer un estudio y consultas sobre las estadísticas que veamos oportunas.

- **Datos a procesar:**

El analizador léxico de Lex tendrá que reconocer los siguientes datos de cada lista a estudiar.

1. Sobre la lista de premios tendrá que reconocer el boleto o boletos premiados junto con el tipo de premio, que estará en un intervalo de 1º premio a 5º premio. El número de boletos premiados en cada intervalo en principio puede variar, dependiendo del sorteo (si hay más de un primer premio...)
2. Sobre las administraciones se reconocerán: un identificador, el nombre de las mismas, la dirección, la ciudad o pueblo junto con su provincia, el número de teléfono de la empresa y el enlace web. Cada administración, tendrá una lista de los boletos vendidos (diferenciados de los premiados porque van entre “-“).
3. Finalmente los participantes se reconocerán por DNI y boletos comprados (diferenciados del resto porque van entre “+“).

Las expresiones regulares utilizadas para cada caso, se especifican en el fichero `loteria.l` con su alias correspondiente.

- **Estructuras de datos:**

Para almacenar internamente los datos una vez reconocidos he usado vectores dinámicos y estructuras tipo `struct`. Debido a que no hemos trabajado mucho en C, he tenido que usar algunos métodos extra para garantizar el almacenamiento de cada conjunto de datos.

1. La lista de boletos premiados la he almacenado en un vector de enteros de la siguiente forma. Número de premio `[-5,-1]` seguida de una lista de boletos asignados a ese premio.

Ejemplo: `-1 72671 -2 74838 -3 83828 -4 84738 47738 47382...`

Los tipos de premios los pongo en negativo para diferenciarlos de números del sorteo como `00001` ya que internamente ambos serían tratados como `1`.

Utilizando una función auxiliar que te devuelve el intervalo de datos en los que se encuentran el boleto o boletos de ese premio.

2. Las administraciones se almacenan en un array de `struct Administracion` que contiene un campo para cada dato. Para que el programa sepa cuándo debe pasar de administración en el array de `struct`, hay que imponer que la web (podría ser cualquier otro dato), sea siempre el último dato entre una administración y otra en el fichero. El resto de datos se reconocen en cualquier orden.

La adjudicación de los boletos se hace más tarde, ya que los boletos vendidos por cada administración, se guardan en un array junto con el identificador de la administración, de forma similar que se hacía en los premios. Esto no es más que para evitar código confuso en las reglas de producción.

3. Finalmente los participantes también se almacenan en un array de `struct Participante` que contiene un campo para el DNI y otro para los boletos comprados.

En este caso también se adjudican los boletos a cada comprador más tarde por el mismo motivo que las administraciones. Usando un array de

enteros extra donde se guardan DNI (en forma entero) junto con sus boletos.

- **Tratamiento de los datos:**

Finalmente, una vez que tenemos todo el texto de fichero procesado y almacenado, solo nos queda utilizarlo. Podría hacer infinidad de estudios sobre los datos, pero me he centrado en cuatro:

1. Identificador de boletos premiados por participante. Sobre los DNI almacenados, hace un estudio de los premios que ha ganado cada uno.
2. Total de premios que ha vendido cada localidad.
3. Detector de boletos premiados, introducir un número desde pantalla y decirte si tiene o no premio.
4. Muestra las administraciones de lotería junto con sus principales datos.

El código además está comentado, por si hay algún detalle que se me haya podido olvidar en este documento, sobre el funcionamiento de algún método o variable usada.