## Práctica p3b extraordinaria

La gramática siguiente genera un subconjunto del lenguaje de entrada para `dot`, que describe grafos a base de una lista ($L$) de caminos ($C$) entre nodos:

$$G_1 = \begin{cases} F \rightarrow \textbf{graph id}\,\{\,L\,\} \\ L \rightarrow L;C \mid C \\ C \rightarrow N \mid N\ E \\ N \rightarrow \textbf{id} \mid \textbf{longId} \\ E \rightarrow \textbf{edge}\ N\ E \mid \textbf{edge}\ N \end{cases}$$

donde el terminal **graph** corresponde al lexema "`graph`", **edge** al lexema "`--`", **id** a cualquier cadena alfanúmerica (alfabeto inglés) que no comience con un dígito, pudiendo también incluir el carácter de barra baja (_) en cualquier posición, y **longId** a cualquier cadena entrecomillada que no contenga avances de línea, pero pueda contener comillas precedidas de la barra invertida (\).

Puede haber espacios en blanco, y avances de línea separando componentes léxicos.

Por ejemplo,

```
graph miGrafo { NodoAislado ; Nodo_Raiz -- Nodo -- "Nodo \"final\""; Nodo -- Otro }
```

sería una cadena de entrada correcta, que utiliza tres "caminos" para describir un grafo de 5 nodos y 3 arcos y que puede verse en la figura de más abajo.

Realice un analizador sintáctico descendente recursivo (en C), con un analizador léxico adecuado, para obtener

1. un archivo de texto en el que aparezcan tantas líneas como nodos. En cada línea, un número de orden, el nombre del nodo, –entrecomillado si no lo estaba en el fichero fuente–, el número de sus hijos, y el nombre de su nodo "padre".

2. un archivo de texto en el que aparezca una matriz bidimensional de unos y ceros, representando si hay o no arco entre las coordenadas: 1 si hay arco entre el nodo de la fila y columna correspondiente, 0 si no lo hay.

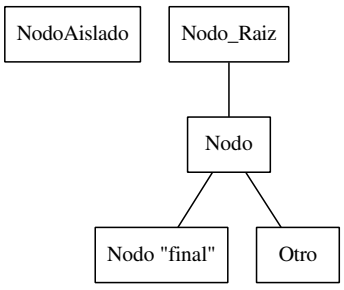
Suponga que el grafo descrito es un árbol (de forma que cada nodo tiene a lo sumo un padre) y que cada arco aparece descrito solamente una vez.

Para el ejemplo, el primer fichero de salida podría ser el siguiente:

```
0 "NodoAislado" 0
1 "Nodo_Raiz" 1
2 "Nodo" 2 "Nodo_Raiz"
3 "Nodo \"final\"" 0 "Nodo"
4 "Otro" 0 "Nodo"
```

y el segundo debe ser:

```
0 0 0 0 0
0 0 1 0 0
0 0 0 1 1
0 0 0 0 0
0 0 0 0 0
```




Puede modificarse la gramática (o utilizar otra) si se considera necesario, pero manteniendo la equivalencia con la propuesta.

Documente el programa con un documento externo en el que se muestren las modificaciones realizadas en la gramática (o la nueva gramática si procede), y su motivo, la descripción de los componentes léxicos, las tablas necesarias para el análisis sintáctico y el esquema de traducción implantado.