Parseador de Topologías a archivos .TOP(Realizado en C , No compatible para linux)

**Input:** Directorio que albergue directorios a su vez de topologías a parsear, este debe incluirse en el directorio donde se encuentre él programa. Toma el archivo *RyuFileEdges.txt* de cada topología para conocer el numero de enlaces y que nodos conforman cada enlace. Si se modifica el nombre del fichero *RyuFileEdges.txt* habría que modificar a su vez la string constante que lo contiene (línea 246 dentro de la función int Parser(char \*, char \*);).

**Output:** Genera archivos parseados del tipo (“nombre\_de\_la\_topologia.top”) dentro de cada directorio de cada topología.

**Funcionamiento:** El programa pregunta por el nombre del directorio raíz de las distintas topologías, por defecto usará *“Topologias\_Brite”*. Cuando obtiene el nombre del directorio raíz, hace un barrido para contar todos los directorios disponibles dentro del directorio raíz.

Sabiendo ya al número de topologías que va a tener que parsear, hará una reserva dinámica de memoria para un array bidimensional que guardará los nombres de los distintos directorios que albergan topologías.

Sabiendo ya las distintas rutas a cada archivo *RyuFileEdges.txt* de cada topología, se empieza a parsear, haciendo uso de las funciones:

- int Parser(char \*, char \*);

Abre y crea el archivo final “.top” . Y final mente se encarga de ir completando el nuevo archivo *nombre\_de\_la\_topologia.top* .

- int read(FILE \*, int \*, int \*);

Lee el archivo *RyuFileEdges.txt* y obtiene información de que nodos componen cada enlace.

- int NumEnlaces(FILE \*);

Lee el archivo *RyuFileEdges.txt*  y cuenta el número de líneas que tiene este archivo, que su vez serán el número de enlaces que tenga esa topología(En el return resta una línea ya que por defecto los archivos *RyuFileEdges.txt* contienen un salto de línea al final de cada archivo)

Una vez parseado un archivo del tipo *RyuFileEdges.txt* se cierran con fclose, y continúan con la siguiente topología.

Cuando se parsean todas las topologías, se libera la memoria reservada dinámicamente anteriormente para albergar el nombre de cada topología.