MEMORIA DINÁMICA:

1. El usuario teclea un número indeterminado de frases por teclado, tenemos que almacenarlas en memoria dinámica, tanto en número de frases como la longitud de las mismas. Cuando el usuario teclea un ***** el programa termina:
    a. Muestra un listado de todas las frases tecleadas.
    b. Después hay que liberar toda la memoria ocupada.

2. Implementar un vector dinámico, se puede rellenar de números aleatorios.

3. Implementar una matriz dinámica, y funciones que permitan imprimir, sumar y multiplicar matrices.

ESTRUCTURAS DINÁMICAS:

4. Implementar un menú que muestre lo siguiente:
   MENU PRINCIPAL
   1.- Insertar datos.
   2.- Listar cuentas.
   3.- Buscar cuenta.
   4.- Borrar cuenta.
   5.- Borrar todas las cuentas.
   6.- Salir

   - Se mantendrá la información en una **lista lineal.**
   - La parte de los datos contendrá los siguientes campos:
   char *nroCuenta, char *DNI, float saldo.
   - Los prototipos de todas las funciones, irán en el fichero: prototipos.h
   - La definición de estructuras ira en el fichero: estructuras.h
   - Main.c se implementarán las funciones:
      - main() → se encarga de mostrar el menú y gestionar las opciones.
      - menu() → Muestra el menú por pantalla.
      - leerCuenta() → Lee los datos de la cuenta de teclado.
      - imprimeCuenta() → Muestra los datos de la cuenta por pantalla.

- listarCuentas() → Lista todas las cuentas.
- insertarDatos() → Llama a leer cuenta y luego añade la cuenta a la lista.
- añadeCuenta() → Función recursiva que añade los datos de la cuenta en la lista, al final.
- buscarCuenta() → Pide un número de cuenta por pantalla y llama a otra función recursiva que devolverá NULL si no encuentra la cuenta y si no mostrará los datos de la misma.
- borrarCuenta() → Idem de la anterior pero borrando, pero no es recursiva.
- borrarCuentas() → Libera la memoria de todas las cuentas.

5. Implementar un menú para gestionar un **árbol binario de búsqueda**:
   MENÚ PRINCIPAL
   1.- Añadir un número.
   2.- Los 3 recorridos.
   3.- Buscar un elemento.
   4.- Liberar todos los nodos.
   5.- Calcular la altura.
   6.- Salir.
   Los elementos del árbol pueden ser **int**. Seguir los mismos criterios de estructurar los ficheros que en la lista lineal.

6. Implementar una **cola**, con las funciones de encolar (añadir un nuevo elemento) y la función desencolar (sacar un elemento). Será de números mantendremos dos punteros uno a la cabecera y otro al final. Los números se cargarán a partir de un fichero de texto, en el que cada número vendrá en una línea. Se encolarán todos los números y luego se desencolaran imprimiendo por pantalla.

7. En una **lista** hay que implementar las siguientes operaciones:
   a. Insertar primero.
   b. Insertar último (iterativo y recursivo).
   c. Insertar ordenado.
   d. Buscar un elemento (por dato o por posición).
   e. Borrar un elemento (por dato o por posición).

f. Modificar un elemento (algún dato).
g. Liberar todos los elementos (iterativo / recursivo).
h. Contar los elementos.
i. Imprimir los elementos.
j. Especificar todas las operaciones con un menú.