

Práctica 3

GoT: Gestión de Personajes de Game of Thrones

Algoritmos y Estructuras de Datos - Grado en Ingeniería Informática (USC)

9 de octubre de 2024

Objetivos

En esta práctica trataremos de:

- Reutilizar los TADs conocidos (árbol binario de búsqueda, lista, cola).
- Valorar qué TAD es el más adecuado para el diseño de las diferentes tareas de un programa.

1. Versión 1 de la práctica

En esta primera versión de la práctica utilizaremos el TAD árbol binario de búsqueda para guardar los datos de un conjunto de personajes. Los datos que tendremos para cada personaje son: name, house, royal y varias listas: parents, siblings y killed.

El programa se iniciará creando la base de datos de personajes vacía. Para la base de datos se utilizará un TAD árbol binario de búsqueda que pueda almacenar tantos personajes como sea necesario. A continuación se ofrecerá el menú de selección de opciones que se repetirá continuamente hasta que el usuario elija la opción de salir. Por último, se destruirá la base de datos. En el menú se mostrarán las siguientes opciones:

- **A/a. Añadir personaje:** esta opción permitirá añadir un nuevo personaje a la base de datos. Al añadir un nuevo personaje se preguntará su nombre, su casa, si es de familia real y se rellenarán varias listas: padres, hermanos, y personas que ha asesinado. Para cada lista se preguntará el nombre del personaje, finalizando el proceso de rellenarla cuando el nombre del personaje sea "fin".
- **L/l. Listado alfabético de personajes:** Se imprimirán los datos de los personajes de la base de datos en orden alfabético. También se imprimirá de forma detallada la información de **todas** sus características.
- **E/e. Eliminar personaje:** esta opción permitirá eliminar un personaje de la base de datos. Se preguntará su nombre y, si está en la base de datos, se informará de su baja. Si no está en la base de datos, también se informará por pantalla.
- **S/s. Salir:** esta opción sale del menú, finalizando el programa.

Temporización recomendada

A continuación se presenta una temporización recomendada para realizar cada una de las fases del proyecto. Tened en cuenta que esto es sólo una estrategia de implementación: tenéis completa libertad para implementar las funcionalidades que se piden en el enunciado, siempre y cuando utilicéis el TAD abb y el TAD lista.

Fase 1: Preparación del proyecto.

En primer lugar, debéis crear un proyecto con el nombre `juegodetronos` y crear `main.c`. Todas las funciones que se llaman desde el menú principal deben estar implementadas en un archivo `juegodetronos.c` con su correspondiente interfaz `juegodetronos.h`, que iremos ampliando en las distintas versiones del programa.

Fase 2: Preparación de los TAD

El programa deberá implementar obligatoriamente el árbol y las listas indicadas con los TAD correspondientes, cuya descripción y funcionalidades se han explicado en clase de teoría. Antes de iniciar el desarrollo de la práctica debéis repasar detalladamente las diapositivas y notas de teoría de los temas correspondientes y, de forma especial, los ejemplos, para recordar el funcionamiento de los TAD.

NO PODRÁ REALIZARSE NINGUNA MODIFICACIÓN SOBRE LOS FICHEROS DE LOS TAD PROPORCIONADOS excepto las indicadas a continuación, además de las que necesitéis para añadir los `#include` necesarios para modificar los tipos de elemento de los TAD, así como definir constantes para las longitudes de las cadenas de caracteres.

Preparación del TAD abb (de personajes)

Para la base de datos de personajes se utilizará el TAD abb implementado en los ficheros adjuntos (`abb.c` y `abb.h`) en el que `TIPOELEMENTOABB` debe ser una estructura con la información del personaje: campos `name` y `house` (podéis usar un array estático definiendo su longitud como una constante), campo `royal` (tendrá un entero con valor 0 o 1), y varias `TLISTA` para almacenar los campos `parents`, `siblings` y `killed`. También debéis modificar `TIPOCLAVE` para indicar cuál va a ser vuestro campo de ordenación en el abb, si ordenamos por orden alfabético será un `char *`.

Además, debéis modificar las funciones `_clave_elem()`, `_comparar_claves()` y `_destruir_elem()` en `abb.c` para adaptarlas a vuestro `TIPOELEMENTOABB`. Recordad que para usar `strcmp()` debéis incluir la librería `string.h`.

Preparación del TAD lista (de parents, siblings y killed)

Para las listas de `parents`, `siblings` y `killed` se utilizará el TAD Lista implementado en los ficheros adjuntos (`lista.c` y `lista.h`) en el que el `TIPOELEMENTOLISTA` debe definirse como una estructura con el campo `nameP` (podéis usar un array estático, pero entonces debéis definir la longitud como una constante).

Fase 3: Implementación del menú

Por último, deberéis implementar el menú del programa. El menú debe funcionar con letras mayúsculas o minúsculas. Las opciones del menú que debería haber implementadas son las siguientes:

- **A/a. Añadir personaje:** tendréis que llamar a la función `anhadirPersonaje()` que realiza la toma de datos por teclado de la información del personaje a añadir a la base

de datos. Tenéis que tener en cuenta que debéis implementar un bucle para poder tomar información sobre las personas de las distintas listas mientras el nombre de la misma sea distinta de “fin” (puede resultar útil aquí la función `strcmp()` incluida en la librería `string.h`).

- **L/l. Listado alfabético de personajes:** Tenéis que tener en cuenta que las funciones para imprimir podéis implementarlas en cadena, es decir, `listadoPersonajes()` llamaría a `_imprimirPersonaje()` e `_imprimirPersonaje()` llamaría varias veces a `_imprimirLista()`.
- **E/e. Eliminar personaje:** tendréis que llamar a la función `eliminarPersonaje()` que solicitará el nombre del personaje a eliminar y, si está en la base de datos, procederá a eliminarlo, informando por pantalla del resultado de la acción.
- **S/s. Salir:** tendrá como efecto salir del bucle del menú.

Una vez salimos del menú, habrá que proceder con la destrucción de la base de datos de memoria.

Ejemplo de ejecución versión 1

A continuación se muestra un ejemplo de ejecución del programa. Tened en cuenta que el formato de vuestro menú puede diferir de lo aquí expuesto.

```
1 > ./juegodetronos
2
3 Bienvenid@ al programa de gestion de personajes GoT
4 A. Anhadir personaje
5 L. Listar personajes
6 E. Eliminar personaje
7 S. Salir
8 Seleccione una opcion: A
9
10 Name: Aegon Targaryen
11   House (? si desconocido): Targaryen
12   Royal (0/1): 1
13   Parents (fin para terminar): Elia Martell
14   Parents (fin para terminar): Rhaegar Targaryen
15   Parents (fin para terminar): fin
16   Siblings (fin para terminar): Rhaenys Targaryen
17   Siblings (fin para terminar): Jon Snow
18   Siblings (fin para terminar): fin
19   killed (fin para terminar): fin
20
21 A. Anhadir personaje
22 L. Listar personajes
23 E. Eliminar personaje
24 S. Salir
25 Seleccione una opcion: a
26
27 Name: Aeron Greyjoy
28   House (? si desconocido): Greyjoy
29   Royal (0/1): 0
30   Parents (fin para terminar): fin
31   Siblings (fin para terminar): Balon Greyjoy
32   Siblings (fin para terminar): Euron Greyjoy
33   Siblings (fin para terminar): fin
34   killed (fin para terminar): fin
35
```

```

36 A. Anhadir personaje
37 L. Listar personajes
38 E. Eliminar personaje
39 S. Salir
40 Seleccione una opcion: A
41
42 Name: Aerys II Targaryen
43   House (? si desconocido): Targaryen
44   Royal (0/1): 1
45   Parents (fin para terminar): fin
46   Siblings (fin para terminar): Rhaella Targaryen
47   Siblings (fin para terminar): fin
48   killed (fin para terminar): Brandon Stark
49   killed (fin para terminar): Rickard Stark
50   killed (fin para terminar): fin
51
52 A. Anhadir personaje
53 L. Listar personajes
54 E. Eliminar personaje
55 S. Salir
56 Seleccione una opcion: L
57
58 Aegon Targaryen
59   House: Targaryen
60   Royal: Yes
61   Parents: Elia Martell, Rhaegar Targaryan
62   Siblings: Rhaenys Targaryan, Jon Snow
63
64 Aeron Greyjoy
65   House: Greyjoy
66   Siblings: Balon Greyjoy, Euron Greyjoy
67
68 Aerys III Targaryen
69   House: Targaryen
70   Royal: Yes
71   Siblings: Rhaella Targaryen
72   killed: Brandon Stark, Rickard Stark
73
74 A. Anhadir personaje
75 L. Listar personajes
76 E. Eliminar personaje
77 S. Salir
78 Seleccione una opcion: E
79
80 Nombre del personaje a eliminar: Aeron Greyjoy
81 El personaje Aeron Greyjoy ha sido eliminado de la base de datos
82
83 A. Anhadir personaje
84 L. Listar personajes
85 E. Eliminar personaje
86 S. Salir
87 Seleccione una opcion: E
88
89 Nombre del personaje a eliminar: Daenerys Targaryen
90 Este personaje no esta registrado en la base de datos
91
92 A. Anhadir personaje
93 L. Listar personajes
94 E. Eliminar personaje
95 S. Salir
96 Seleccione una opcion: 1
97
98 Aegon Targaryen

```

```
99      House: Targaryen
100     Royal: Yes
101     Parents: Elia Martell, Rhaegar Targaryan
102     Siblings: Rhaenys Targaryan, Jon Snow
103
104 Aerys III Targaryen
105     House: Targaryen
106     Royal: Yes
107     Siblings: Rhaella Targaryen,
108     killed: Brandon Stark, Rickard Stark
109
110 A. Anhadir personaje
111 L. Listar personajes
112 E. Eliminar personaje
113 S. Salir
114 Seleccione una opcion: s
115
116 Gracias por usar nuestro sistema GoT
117
118 >
```

2. Versión 2: Lectura/Grabación de la base de datos de personajes

En esta versión leeremos los datos de los personajes de un archivo que se introducirá por línea de comandos y también actualizaremos la base de datos guardando los datos actualizados en el mismo archivo.

La lectura del archivo se realizará antes del menú y la grabación se realizará al finalizar el menú, antes de destruir el árbol.

El programa debe funcionar aunque no se introduzca un archivo inicial.

Lectura de datos desde un fichero

Deberá permitirse la lectura de un fichero de personajes con sus datos y la inicialización de la base de datos a partir de este fichero. El fichero deberá tener el siguiente formato: *NAME/HOUSE/ROYAL/PARENT_1,PARENT_2,PARENT_3/SIBLING_1,SIBLING_2,SIBLING_3/KILLED_1,KILLED_2,KILLED_3/*

Un ejemplo se muestra en el siguiente fragmento (disponible para descargar en la tarea del campus virtual):

```
1 Aegon Targaryen|Targaryen|1|Elia Martell,Rhaegar Targaryen|Rhaenys
  Targaryen,Jon Snow|-|
2 Aeron Greyjoy|Greyjoy|0|Balon Greyjoy,Euron Greyjoy|-|-|
3 Aerys II Targaryen|Targaryen|1|Rhaella Targaryen|Brandon Stark,Rickard
  Stark|-|
4 Arya Stark|Stark|0|Eddard Stark,Catelyn Stark|Robb Stark,Sansa Stark,Bran
  Stark,Rickon Stark|Polliver,Rorge,Ghita,Meryn Trant,Lothar Frey,Walder
  Frey,Petyr Baelish,The Night King,White Walker,Viserion|
5 Ygritte|?|0|-|-|Old Man|
```

Se asume que el fichero es siempre correcto, para no sobrecargar vuestro proyecto con comprobaciones defensivas.

Pautas sobre la función cargar_archivo()

Esta función debe leer cada línea de un archivo de entrada y almacenar los datos de ese personaje en el árbol con el que representáis la base de datos.

La función `cargar_archivo()` se llama desde `main()` y debe utilizar sus argumentos para comprobar que se ha llamado correctamente desde línea de comandos:

En línea de comandos (cambiad `ejecutable` por el nombre de vuestro programa y `archivo.txt` por el nombre de vuestro archivo de texto):

```
1 ./ejecutable -f archivo.txt
```

En `main()`:

```
1 int main(int argc, char** argv) {
2     TABB ArbolPersonajes;
3     ...
4     cargar_archivo(&ArbolPersonajes,argc,argv);
5     ...
6 }
```

Recordatorio de los argumentos de `main()`:

- `argc` tomará el valor 3: es el número de cadenas que se introducen por la línea de comandos: el ejecutable, la opción `-f` y el nombre del archivo.
- `argv[]` es un array donde se almacenan esas 3 cadenas:
 - `argv[0]` almacena la cadena `./ejecutable`
 - `argv[1]` almacena la cadena `"-f"`
 - `argv[2]` almacena la cadena `"archivo.txt"`

En `juegodetronos.c`: implementamos la función `cargar_archivo()`, que recibirá el árbol donde va a insertar los datos de los personajes que se lean desde fichero. Debe recibir también los argumentos de `main()`, como vimos en el código anterior de `main.c`, para comprobar si existe o no archivo de datos para cargar.

Para poder procesar este fichero debéis tener en cuenta las siguientes consideraciones:

- En primer lugar, debéis procesar los argumentos por línea de comandos (que el número de argumentos sea 3 como mínimo, que el segundo argumento sea `"-f"` y que el tercer argumento sea un nombre de archivo que al intentar abrirlo no de error.
- En segundo lugar, debéis leer el fichero línea a línea. Para ello podéis utilizar la función `fgets()`, que tiene el problema de que añade el retorno de carro a la cadena. Para eliminarlo, podéis utilizar la siguiente función (necesitaréis la librería `string.h`).

```
1 void _strip_line(char * linea){
2     linea[strcspn(linea, "\r\n")] = 0;
3 }
```

- En tercer lugar, debéis procesar los campos de la línea leída. Para eso, necesitaréis dos tipos de llamadas a la función `strtok()`. En el primer tipo de llamada, partiréis la línea a partir del carácter `"|"`. Para eso, es suficiente con hacer tantas llamadas a `strtok()` como campos tenga vuestro personaje, tal y como sigue:

```
1 char * name = strtok(linea, "|"); //primera llamada, se envia la
   cadena linea
2 char * house = strtok(NULL, "|"); //siguientes llamadas, el primer
   argumento es NULL
3 ...
4 char *killed_list = strtok(NULL, "|"); //ultima llamada
```

Después, como las funciones que usan `strtok()` sobrescriben la cadena original, no podemos usarla sobre las cadenas que tienen la lista de parents, siblings y killed. Esto se puede hacer con la función `strtok_r()` o `strtok_s()` que no están disponibles en todos los entornos (por ejemplo, en Windows con cygwin no están disponibles). Por tanto, escribiremos nuestra función `_splitByComma()` para dividir cada una de estas listas en elementos separados por comas y así insertarlos en la lista que corresponda.

A continuación tenéis un ejemplo de uso de `strtok()`:

```
1 char *cadena="Hola,mundo,soy,una,cadena,separada,por,comas";
2 char *token=strtok(cadena, ",");
3 while(token!=NULL){
4     printf("Token:%s\n", token);
5     token=strtok(NULL, ",");
6 }
```

A continuación tenéis un ejemplo de una función que hace lo mismo escrita por nosotros:

```
1 char *cadena="Hola ,mundo ,soy ,una ,cadena ,separada ,por ,comas ";
2 char *start=cadena;
3 char *end=cadena;
4
5 while(*end!='\0'){
6     if (*end==',' ){
7         *end='\0';
8         printf("Token: %s\n",start);
9         start=end+1;
10    }
11    end++;
12 }
13 printf("Token: %s\n",start); //Print the last token
```

Tenéis total libertad para utilizar alguna de estas funciones o generar vosotros una función que realice la lectura de los personajes, sólo tened en cuenta que strtok() no se puede usar sobre dos cadenas iniciales simultáneamente, es decir, no la podemos usar al mismo tiempo para separar las cadenas separadas por “|” y las cadenas separadas por “,”.

Grabación de datos en un fichero

El fichero debe actualizarse con los últimos datos disponibles por lo que, si tenéis un fichero precargado que habéis introducido por línea de comandos, debéis actualizarlo abriéndolo para escritura y escribiendo los datos del árbol de personajes. Tened cuidado con el recorrido que hacéis para escribir, pues si elegís un recorrido inorden los datos se guardarán en el archivo por orden alfabético y por tanto la siguiente vez que los leamos el abb estará totalmente desequilibrado. El fichero de salida debe tener el mismo formato que el fichero de entrada, para poder leerse posteriormente. En caso de que no hayáis precargado un archivo por línea de comandos, debéis preguntar al usuario el nombre del fichero de salida con el que quiere guardar la base de datos, y así servirá para siguientes ejecuciones.

Ejemplo de ejecución versión 2 (lectura/escritura de archivos)

```
1 > cat personajes.txt
2 Aegon Targaryen|Targaryen|1|Elia Martell ,Rhaegar Targaryen|Rhaenys
   Targaryen ,Jon Snow|-|
3 Aeron Greyjoy|Greyjoy|0|Balon Greyjoy ,Euron Greyjoy|-|-|
4 Aerys II Targaryen|Targaryen|1|Rhaella Targaryen|Brandon Stark , Rickard
   Stark|-|
5 Arya Stark|Stark|0|Eddard Stark ,Catelyn Stark|Robb Stark ,Sansa Stark , Bran
   Stark ,Rickon Stark|Polliver ,Rorge ,Ghita ,Meryn Trant ,Lothar Frey ,Walder
   Frey ,Petyr Baelish ,The Night King ,White Walker ,Viserion|
6 Ygritte|?|0|-|-|Old Man|
7
8 > ./juegodetronos -f personajes.txt
9
10 Bienvenid@ al programa de gestion de personajes GoT
11 A. Anhadir personaje
12 L. Listar personajes
13 E. Eliminar personaje
14 S. Salir
15 Seleccione una opcion: L
16
```



```

17 Name: Aegon Targaryen
18   House: Targaryen
19   Royal: Yes
20   Parents: Elia Martell, Rhaegar Targaryen
21   Siblings: Rhaenys Targaryen, Jon Snow
22
23 Name: Aeron Greyjoy
24   House: Greyjoy
25   Parents: Balon Greyjoy, Euron Greyjoy
26
27 Name: Aerys II Targaryen
28   House: Targaryen
29   Royal: Yes
30   Parents: Rhaella Targaryen
31   Siblings: Brandon Stark, Rickard Stark
32
33 Name: Arya Stark
34   House: Stark
35   Parents: Eddard Stark, Catelyn Stark
36   Siblings: Robb Stark, Sansa Stark, Bran Stark, Rickon Stark
37   killed: Polliver, Rorge, Ghita, Meryn Trant, Lothar Frey, Walder Frey,
      Petyr Baelish, The Night King, White Walker, Viserion
38
39 Name: Ygritte
40   killed: Old Man
41
42 A. Anhadir personaje
43 L. Listar personajes
44 E. Eliminar personaje
45 S. Salir
46 Seleccione una opcion: A
47
48 Nombre del personaje: Alton Lannister
49   House (? si desconocido): Lannister
50   Royal (0/1): 0
51   Parents (fin para terminar): fin
52   Siblings (fin para terminar): fin
53   Killed (fin para terminar): fin
54
55 A. Anhadir personaje
56 L. Listar personajes
57 E. Eliminar personaje
58 S. Salir
59 Seleccione una opcion: S
60
61 Gracias por usar nuestro sistema GoT
62
63 >
64 > cat personajes.txt
65 Aegon Targaryen|Targaryen|1|Elia Martell,Rhaegar Targaryen|Rhaenys
      Targaryen,Jon Snow|-|
66 Aeron Greyjoy|Greyjoy|0|Balon Greyjoy,Euron Greyjoy|-|-|
67 Aerys II Targaryen|Targaryen|1|Rhaella Targaryen|Brandon Stark, Rickard
      Stark|-|
68 Arya Stark|Stark|0|Eddard Stark,Catelyn Stark|Robb Stark,Sansa Stark, Bran
      Stark,Rickon Stark|Polliver,Rorge,Ghita,Meryn Trant,Lothar Frey,Walder
      Frey,Petyr Baelish,The Night King,White Walker,Viserion|
69 Ygritte|?|0|-|-|Old Man|
70 Alton Lannister|Lannister|0|-|-|-|
71
72 >

```

3. Versión 3: Búsqueda de información

Partiremos del fichero `personajes.txt`, que contiene casi todos los personajes de GoT, y buscaremos información en este árbol.

El menú de la aplicación ahora tendrá tres funcionalidades más, que debéis realizar de la forma más modular posible:

- **B/b. Buscar asesino:** tendréis que llamar a la función `buscarAsesino()` que solicita por teclado el nombre del personaje que ha sido asesinado (que puede pertenecer o no a la base de datos) y lo busca en la lista `killed` de todos los personajes de la base de datos.
- **H/h. Buscar hijos/as:** tendréis que llamar a la función `buscarHijos()` que solicita por teclado el nombre de un personaje (que puede pertenecer o no a la base de datos) y lo busca en la lista `parents` de todos los personajes de la base de datos.
- **K/k. Buscar killer:** En este caso debéis buscar el personaje que más asesinatos ha realizado.
- **M/m. Modificar personaje:** Debéis preguntar el nombre del personaje y, si está en la base de datos, ofrecer un menú para elegir el campo a modificar. Se pueden modificar todos los campos, pues puede haber alguna errata en el nombre, podemos tener nueva información sobre su pertenencia a una casa (por ejemplo, porque se ha descubierto que uno de sus padre pertenece a una casa o por matrimonio), ha podido convertirse en miembro de la realeza, y la lista de padres, hermanos y personas asesinadas ha podido incrementarse (en este caso se añadirán los nuevos personajes a cada lista).

Ejemplo de ejecución versión 3

```
1 > ./juegodetronos -f personajes.txt
2
3 Bienvenid@ al programa de gestion de personajes de Juego de Tronos
4 A. ñAadir personaje
5 L. Listar personajes
6 E. Eliminar personaje
7 B. Buscar asesino
8 H. Buscar hijos
9 K. Buscar killer
10 M. Modificar personaje
11 S. Salir
12
13 Seleccione una opcion: b
14
15 Nombre de personaje asesinado: Jon Snow
16 Asesinos/as: Alliser Thorne, Bowen Marsh, Olly, Othell Yarwyck
17
18 A. ñAadir personaje
19 L. Listar personajes
20 E. Eliminar personaje
21 B. Buscar asesino
22 H. Buscar hijos
23 K. Buscar killer
24 M. Modificar personaje
25 S. Salir
26
27 Seleccione una opcion: b
```

```

28
29 Nombre de personaje asesinado: Daenerys Targaryan
30     Asesinos/as: 0 no áest muerto o su asesino no áest en la base de
      datos
31
32 A. ñAadir personaje
33 L. Listar personajes
34 E. Eliminar personaje
35 B. Buscar asesino
36 H. Buscar hijos
37 K. Buscar killer
38 M. Modificar personaje
39 S. Salir
40
41 Seleccione una opcion: h
42
43 Nombre de personaje del que quieres buscar sus hijos: Elia Martell
44     Hijos/as: Aegon Targaryen, Rhaenys Targaryen
45
46 A. ñAadir personaje
47 L. Listar personajes
48 E. Eliminar personaje
49 B. Buscar asesino
50 H. Buscar hijos
51 K. Buscar killer
52 M. Modificar personaje
53 S. Salir
54
55 Seleccione una opcion: h
56
57 Nombre de personaje del que quieres buscar sus hijos: Robb Stark
58     Hijos/as: Si tiene hijos, estos no figuran en la base de datos
59
60 A. ñAadir personaje
61 L. Listar personajes
62 E. Eliminar personaje
63 B. Buscar asesino
64 H. Buscar hijos
65 K. Buscar killer
66 M. Modificar personaje
67 S. Salir
68
69 Seleccione una óopcion: k
70     Los killers, con 16 asesinatos son: Daenerys Targaryen, Sandor
      Clegane
71
72 A. ñAadir personaje
73 L. Listar personajes
74 E. Eliminar personaje
75 B. Buscar asesino
76 H. Buscar hijos
77 K. Buscar killer
78 M. Modificar personaje
79 S. Salir
80
81 Seleccione una óopcion: s
82
83 Gracias por usar nuestro sistema GoT
84 >

```

Entregables y calificaciones

Deberá realizarse la entrega por el Campus Virtual, donde están especificadas las fechas de entrega particulares para cada grupo y los criterios de evaluación. Las instrucciones para generar el fichero para su entrega son las siguientes:

- Debéis subir un único fichero comprimido en zip con el nombre (sin tildes ni eñes) `Apellido1Apellido2Nombre_3.zip`.
- Se incluirán SIN SUBCARPETAS todos los ficheros `.c` y `.h` necesarios para compilar el proyecto junto con los archivos de los TADs utilizados. Se asumirá que todos los archivos necesarios para compilar están en la misma carpeta. Se recomienda adjuntar también el fichero para lectura de datos.

CUALQUIER EJERCICIO QUE NO COMPILE SERÁ EVALUADO CON LA CALIFICACIÓN DE 0. ESTE CRITERIO SE MANTENDRÁ EN EL RESTO DE PRÁCTICAS DE LA ASIGNATURA.