

El Ministerio de Asuntos Extraños ha convocado oposiciones para cubrir **k plazas** y necesita gestionar la **información** personal de los **oposidores** y sus **notas**. Para presentarse a la oposición, los opositores deben dar su identificación. La **oposición** consta de **7 ejercicios** eliminatorios. Esto significa que, **si un opositor no aprueba un ejercicio, no tiene derecho a presentarse a los siguientes**. Una vez **finalizados** todos los ejercicios, **se seleccionan las k mejores notas**.

Para la realización del examen se proporcionan los siguientes ficheros:

- Oposiciones.h y Oposiciones.cpp
- Plantilla_ejerc_2.cpp (Contiene la función main)
- opositores.in y opositores.out

Para gestionar la información, es necesario definir los siguientes tipos estructurados en el **módulo Oposiciones**: (0.5 puntos)

1. **tOpositor**. Guarda la identificación del opositor (cadena de caracteres sin blancos), las notas obtenidas en cada uno de los ejercicios (array estático de enteros de tamaño 7), el número de ejercicios aprobados y la calificación final obtenida (media de las calificaciones de todos los ejercicios).
2. **tListaOpositores**. La lista de los opositores se implementa mediante un **array estático de punteros** a variables de tipo **tOpositor**. El tamaño máximo de la lista es **TAM_MAX = 100**. Se garantiza que nunca habrá mas de **TAM_MAX** opositores.
3. **tListaAprobPlaza**. Representa la lista de los **k** opositores (identificador del opositor) que han aprobado todos los ejercicios y que han obtenido plaza. El **orden de los opositores en la lista es de mayor a menor nota media**. Esta lista está implementada con un **array dinámico** de elementos de tipo **string** (identificador del opositor) de tamaño **k**. El número de plazas **k**, puede ser mayor o menor que el número de aprobados, por lo que puede ocurrir que haya opositores aprobados sin plaza, así como plazas que no lleguen a cubrirse.

En el módulo **Oposiciones** deben implementarse las siguientes funciones:

1. **cargarOpositores**. (1.5 puntos) carga el número de plazas y la información de todos los opositores a partir de los datos del fichero "oposidores.in". Este fichero contiene, como primer dato el número de casos de prueba (2). Cada caso de prueba tiene el siguiente formato:
 - a. Un primer número que representa al número **k** (2 y 5 plazas respectivamente para cada caso de prueba).
 - b. A continuación, se muestra el número total de opositores que se presentan a la oposición (8 y 3 respectivamente).
 - c. Las siguientes líneas contienen el identificador de cada opositor y las calificaciones de dicho opositor. Los identificadores son cadenas de caracteres sin espacios en blanco. Las calificaciones son una secuencia de números hasta

encontrar una nota menor a 5. Si un opositor tiene los 7 ejercicios aprobados entonces hay un centinela 0 para finalizar la búsqueda.

2. `ordenar`. (0.5 puntos) Ordena la lista de opositores por la nota media obtenida en los ejercicios (de mayor a menor).
3. `eliminarOrdenado`. (1.5 puntos) Elimina un opositor de la lista de opositores. Recibe como parámetros la lista de opositores y la posición en que se encuentra el elemento a eliminar.
4. `candidatosConPlaza`. (1.5 punto) Recibe como parámetros la lista de opositores ordenada por la nota media (de mayor a menor), la lista de aprobados y el número de plazas k que se ofertan. Si hay menos aprobados que plazas k , entonces la lista contendrá a todos los aprobados. Por el contrario, si hay más aprobados que número de plazas, entonces la lista contiene los identificadores de los k aprobados con mayor calificación. Los opositores que ya tienen plaza se eliminan de la lista de opositores.
5. `mostrarListaOpositores` y `mostrarAprobadosConPlaza`. (0.5 puntos): muestra ambas listas como se muestra en el ejemplo.
6. `liberarListaOpositores` y `LiberarListaAprobados`. (0.5 puntos): Libera la memoria dinámica de ambas listas.

Programa Principal

Los ficheros que contienen la función `main` y el módulo son dados. En estos ficheros hay que implementar la función `resuelveCaso` y las funciones de pedidas.

La función `resuelveCaso` carga una lista de opositores del fichero "opositores.in", actualizando adecuadamente la información completa de cada opositor. Con los opositores ordenados por su media se obtiene la lista de los opositores aprobados con plaza, los cuales son eliminados de la lista de opositores.



Ficheros proporcionados en el sistema juez

opositores.in	opositores.out
2	Lista con todos los opositores ordenados por calific. media
2	
8	Identificador 11111 - nota final 8.14286
11111	Identificador 55555 - nota final 7
8 9 8 9 7 9 7 0	Identificador 77777 - nota final 5.42857
22222	Identificador 44444 - nota final 5.28571
6 8 9 5 6 7 4	Identificador 22222 - nota final 0
33333	Identificador 33333 - nota final 0
2	Identificador 66666 - nota final 0
44444	Identificador 88888 - nota final 0
5 6 5 5 6 5 5 0	
55555	Número de plazas: 2
9 5 7 5 8 9 6 0	
66666	Lista de opositores que no tienen plaza y suspensos
5 3	
77777	Identificador 77777 - nota final 5.42857
7 5 5 5 5 6 5 0	Identificador 44444 - nota final 5.28571
88888	Identificador 22222 - nota final 0
5 7 4	Identificador 33333 - nota final 0
5	Identificador 66666 - nota final 0
3	Identificador 88888 - nota final 0
22222	
5 6 9 8 6 7 6 0	Lista de opositores que tienen plaza
66666	
8 6 7 5 6 6 5 0	Identificador 11111
77777	Identificador 55555
6 1	Lista con todos los opositores ordenados por calific. media
	Identificador 22222 - nota final 6.71429
	Identificador 66666 - nota final 6.14286
	Identificador 77777 - nota final 0
	Número de plazas: 5
	Lista de opositores que no tienen plaza y suspensos
	Identificador 77777 - nota final 0
	Lista de opositores que tienen plaza
	Identificador 22222
	Identificador 66666