Enunciado Entrega Evaluación Continua 2021-2022 **Estructura de Datos y Algoritmos II**

Ejercicios sobre MEMORIA SECUNDARIA

Ejercicio 1. Organización Directa. Dispersión.

Teniendo en cuenta las especificaciones que se detallan en el **Enunciado 2** de las Prácticas de Memoria Secundaria, correspondientes al Tema 6. Organización de Archivos y el fichero (*alumnos.hash*) que se obtiene como resultado en el método de dispersión, se deben implementar las siguientes funciones:

1. Una nueva función de búsqueda con el siguiente prototipo:

   tipoAlumno *busquedaHash(FILE *f, char *dni, int *nCubo, int *ncuboDes, int *posReg, int *error)

   que busca en el archivo creado un registro a partir de su clave, parámetro *dni* de entrada a la función. Esta función devuelve el registro si lo encuentra o el valor NULL si no existe. Además devuelve información sobre la situación del registro en el fichero en los últimos tres parámetros, información que puede utilizarse en procesos posteriores de modificación y/o eliminación:

   - **nCubo:** número de cubo en el que se encuentra el registro si no está desbordado o en el que debería estar si está desbordado.
   - **nCuboDes:** si el registro está desbordado el número de cubo de desborde en el que se encuentra, considerando los cubos desbordados numerados secuencialmente a continuación de CUBOS. Si el registro no está en el área de desborde se le asigna a este parámetro el valor -1.
   - **posReg:** posición del registro en el cubo en el que se encuentra (inicial o desbordado)
   - **error:** este parámetro sirve para detectar si ha habido algún error en el proceso, la función debe asignarle alguno de los siguientes valores:
     - 0 si el proceso acaba correctamente y el registro exite
     - -1 si el proceso acaba correctamente pero el registro no existe
     - -2 si hay problemas con el fichero de datos
     - -4 si ocurre algún otro error en el proceso

2. Una función que, utilizando la información que aporta la función previa, permita modificar el campo provincia a los registros del fichero que se obtiene como resultado en el método de dispersión (*alumnos.hash*). La función a implementar debe seguir el prototipo:

   int modificarReg(char *fichero, char *dni, char *provincia)

   donde el primer parámetro indica el nombre del fichero *hash*, el segundo el *dni* del alumno a modificar y el último, el nuevo valor del campo provincia que se desea asignar al alumno. La función devuelve un entero con el siguiente valor:

   - el número de cubo en el que se encuentra el registro modificado si existe
   - -1 si el registro no existe
   - -2 si hay problemas con el fichero de datos
   - -4 si ocurre algún otro error en el proceso

Ejercicio 2. Búsqueda Binaria

Para relacionar la información de cada alumno con su expediente académico, se ha generado un fichero indice(*numMat.idx*), asociado al fichero de datos del ejercicio 1(*alumnos.hash*), tomando como entrada el número de matrícula y como salida el DNI de cada alumno. Se trata de un fichero, clasificado por el número de matrícula, que contiene registros de tipoIndice tal como se define en indice.h. Para permitir búsquedas de información a través del número de matricula, se deben codificar las siguientes funciones:

1. Una función que implemente la **búsqueda binaria** sobre este fichero índice según el siguiente prototipo

   int busquedaBinariaIndice(int numMat, char *dni, FILE *f)

   de forma que devuelva, en el segundo parámetro, el valor del campo *dni* asociado a la clave de búsqueda *numMat*, que se proporciona como primer parámetro. La función devuelve un entero con el siguiente valor:

   - 0 si el registro exite
   - -1 si el registro no existe
   - -3 si hay problemas con el fichero índice
   - -4 si ocurre algún otro error en el proceso

2. Una función que, utilizando las funciones codificadas previamente que se necesiten, permita buscar la información correspondiente a un alumno en el fichero hash, a partir de su número de matrícula. El prototipo de la función será:

   tipoAlumno * busqueda(int numMat, char *ficheroDatos,char *ficheroIndice, int *error)

   donde el primer parámetro indica el valor del campo de búsqueda *numMat*, el segundo el nombre del fichero de datos (*alumnos.hash*), el tercero el nombre del fichero indice(*numMat.idx*). Esta función devuelve el registro si lo encuentra o el valor NULL si no existe. El cuarto parámetro permite detectar si ha habido algún error en el proceso, la función debe asignarle alguno de los siguientes valores:

   - 0 si el proceso acaba correctamente y el registro exite
   - -1 si el registro no existe
   - -2 si hay problemas con el fichero de datos
   - -3 si hay problemas con el fichero de índice
   - -4 si ocurre algún otro error en el proceso

Para la realización de la práctica se proporcionan dos ficheros:

- **numMat.idx** fichero indice clasificado por el número de matricula
- **indice.h** con los tipos y prototipos necesarios para el ejercicio 2

Nota Se deben introducir las directivas de compilación necesarias en dispersión.h para evitar errores de compilación

Condiciones de la ENTREGA:

- Se debe subir un **único fichero comprimido** que incluya:
  - un fichero con el código C de las funciones del ejercicio 1 (dispersion.c)
  - un fichero con el código C de las funciones del ejercicio 2 (indice.c)
  - un **fichero de prueba** que permita verificar el correcto funcionamiento de las funciones implementadas
  - un fichero **makefile** que permita la perfecta compilación de estos ficheros junto con los proporcionados en la práctica
- Todos los ficheros deben incluir una línea inicial con el **nombre, dni y grupo de prácticas** del alumno. El fichero comprimido debe subirse a la tarea antes de las **23:55** horas del **13 de Mayo de 2022**. No obstante, la tarea permitirá entregas retrasadas hasta el día **8 de Junio a las 14:00**.