

Hoja 2. FICHEROS

Resolver como ficheros de texto en C++.

1. Diseñar un programa que cree un fichero de números enteros llamado (NUMEROS.DAT) con los números enteros del 1 al 100.
2. Diseñar un programa que cree un fichero con enteros positivos introducidos por teclado (se terminará la introducción de enteros con un 0 que no deberá formar parte del fichero). El nombre del fichero deberá solicitarse por teclado.
3. Diseñar un programa que cuente el número de elementos que hay almacenados en un fichero de reales. El nombre del fichero deberá solicitarse por teclado.
4. Diseñar un programa que calcule la media de los elementos de un fichero de reales.
5. Dado un fichero de números enteros llamado DATOS.DAT, diseñar un programa que cree otro fichero llamado ACUM.DAT que contenga las sumas parciales de los elementos de DATOS.DAT. Ejemplo: si $DATOS.DAT = \{a_1, a_2, \dots, a_n\}$, el fichero de sumas parciales $ACUM.DAT = \{s_1, s_2, \dots, s_n\}$ donde $s_1 = a_1$, $s_2 = a_1 + a_2$, ..., $s_n = a_1 + a_2 + \dots + a_n$.
6. Dado un fichero de números enteros, diseñar un programa que separe los números pares de los impares almacenándolos en dos ficheros: PARES.DAT e IMPARES.DAT.
7. Dado un fichero de números enteros, diseñar un programa que visualice por pantalla los números capicúas que hay en el fichero. Diseñar para ello un subprograma que determine si un número es capicúa.
8. Dados dos ficheros de números enteros de nombres UNO.DAT y DOS.DAT, posiblemente de tamaños diferentes, formar un tercer fichero cuyos elementos resultan de tomar alternativamente un elemento de cada fichero. El nombre del fichero a crear se ha de solicitar por teclado.
9. Dados dos ficheros de números enteros ordenados crecientemente que no contienen elementos repetidos, crear otro fichero también ordenado crecientemente que contenga los elementos de los dos ficheros de partida (y que tampoco tenga elementos repetidos).
10. Dada la siguiente estructura de datos:

tipos

`tEmpleado=registro`

`dni, nombre: cadena`

`horas_trabajadas, precio_hora: entero`

freg

y un fichero de datos de este tipo (cada línea contiene un dato de un empleado, es decir, un campo):

- a) Escribir una acción que liste por pantalla todos los empleados almacenados en el fichero que hayan trabajado más de 40 horas. El listado debe contener el nombre y el sueldo de cada uno de los empleados.
- b) Escribir una acción que muestre por pantalla los datos del empleado con un determinado DNI (pasado como parámetro). Completar la acción anterior añadiendo un parámetro de tipo booleano que represente el hecho de encontrar o no el dato buscado.
- c) Escribir una acción que dado un registro que contiene los datos de un empleado lo añada al final del fichero.
- d) Escribir una acción que dados un registro que contiene los datos de un empleado y una posición pos, inserte el registro en la posición pos del fichero.
- e) Escribir una acción que dado un DNI, elimine del fichero los datos del empleado con ese DNI.

Suponer el fichero **ordenado** de forma creciente por el campo DNI:

- a) Escribir una acción que a partir de un DNI, localice el empleado correspondiente y liste su sueldo.
- b) Acción que dado un registro que contiene los datos de un empleado, lo inserte en el fichero y que éste continúe ordenado.
- c) Escribir una acción para eliminar un dato del fichero. Como el campo DNI es clave, sólo será necesario pasarle como parámetro el DNI del elemento a eliminar.

Resolver como ficheros binarios en C++.

- 11. Resolver el ejercicio anterior suponiendo que el fichero es binario.
- 12. Se quiere llevar el control de los suscriptores de una revista. Para esto, se utiliza un fichero que contiene los datos de éstos, en registros del tipo

tipos

tSuscriptor=**registro**

idS:**entero**

nombre, direccion, telef, ciudad:**cadena**

freg

- a) Diseñar un subprograma para crear el fichero leyendo los datos desde el teclado, hasta leer un dato con el campo idS igual a 0. Se aconseja utilizar un subprograma para leer los datos de cada suscriptor.
 - b) Diseñar un subprograma para añadir una nueva suscripción al fichero.
 - c) A partir del fichero de suscripciones y bajo la hipótesis de que hay al menos un suscriptor y que ningún suscriptor aparece dos veces (suponer que idS identifica a los suscriptores), escribir un subprograma que permita modificar el teléfono de un suscriptor dado.
 - d) Diseñar un subprograma para crear el fichero leyendo los datos desde el teclado (terminar al leer un dato con el campo idS igual a 0), ordenado crecientemente por num.
-
- 13. Los promotores de la vuelta ciclista a España, quieren crear una base de datos para controlar a los participantes. La información que se quiere almacenar de cada ciclista es el nombre, dorsal, equipo y la posición en la clasificación, y para ello se va a utilizar un fichero llamado GENERAL.DAT.
 - a) Construir las estructuras de datos que se necesita para almacenar los datos de un ciclista.
 - b) Construir una acción para introducir (leer) los datos de un ciclista y otra acción para mostrarlos (escribirlos). Diseñar también un subprograma que cree un fichero con los datos de todos los ciclistas. (Consideraremos que la introducción de datos termina al introducir un dorsal 0).
 - c) Construir una acción que permita escribir por pantalla los datos correspondientes a los ciclistas de un equipo dado (que será un parámetro de la acción).

14. Supongamos el siguiente entorno de trabajo:

```
tipos
    tJugador=registro
        nombre: cadena
        edad:entero
        nombre_equipo:cadena
        dinero_ofrecido:entero
freg
```

En un fichero de datos de tipo tJugador, ordenado de mayor a menor por el campo dinero_ofrecido, un jugador puede aparecer varias veces porque varios equipos están interesados en su contratación. Se supone que no puede haber dos equipos distintos que ofrezcan la misma cantidad de dinero a un mismo jugador. Escribir una acción con la siguiente cabecera:

```
acción borra_duplicados (ent nom_fich: cadena ; ent nom_jug: cadena)
```

que a partir del fichero de nombre nom_fich construya otro fichero de datos de tipo tjugador llamado FICHAJES.DAT tal que:

- del jugador nom_jug sólo aparezca el registro que contenga el equipo que más dinero ofrece a dicho jugador.
- del resto de jugadores aparezcan los registros de todos los equipos interesados en su fichaje.

15. Supongamos el siguiente entorno de trabajo:

```
talumno=registro
    nombre:cadena
    num_matricula:entero
    programacion1, programacion2:booleano
freg
```

Los datos correspondientes a los alumnos de primer curso de ITIG y Matemáticas se encuentran en el fichero ALUMNOS.DAT cuyos datos son de tipo talumno. Se sabe que no todos los alumnos están matriculados en las dos asignaturas, y que el fichero no está ordenado por ningún campo. Escribir una acción que a partir del fichero anterior construya otro que contenga sólo los alumnos que estén matriculados en las dos asignaturas. Esta misma acción debe mostrar por pantalla el número de alumnos matriculados en cada asignatura.

16. Una empresa especializada en la venta de películas de vídeo dispone de un fichero de registros llamado PELIS.DAT con información sobre las cintas de las que dispone. En él se almacenan para cada película su título, su categoría (comedia, acción, aventuras,...), el precio de venta y el número de ejemplares del que se dispone para su venta. Para ello se dispone del siguiente tipo de datos:

```
tPeli=registro
    titulo,categoría:cadena
    precio:real
    num_ejemp:entero
freg
```

La empresa recibe pedidos de comercios a los que suministra las películas. Para realizar un pedido, un comercio debe indicar el título de la película y el número de ejemplares que se desean recibir.

- a) Escribir un subprograma que a partir del nombre del fichero y de los datos de un pedido, recoja en el fichero la modificación correspondiente al satisfacer el pedido (en caso de que el pedido pueda satisfacerse) y deje el fichero como está en caso contrario (suponer que el fichero está ordenado alfabéticamente por categorías y dentro de cada categoría por orden alfabético de títulos).
- b) Escribir un programa que utilice el subprograma anterior.

17. Una agencia de alquiler de coches dispone de un fichero de registros llamado COCHES.DAT con información sobre sus coches. En él se almacenan la marca (Opel, SEAT,...), la matrícula y kilómetros recorridos por la flota de coches de la agencia. Suponer que el fichero está ordenado alfabéticamente por marcas. Para ello se dispone del siguiente tipo de datos:

```
tcoche=registro
      marca,matricula:cadena
      kms:real
freg
```

Escribir un subprograma que a partir de los datos de un coche (parámetro *ch* de tipo *tcoche*) y el nombre del fichero físico, actualice el contenido del fichero COCHES.DAT con los datos almacenados en *ch*. Notar que la matrícula almacenada en *ch* puede corresponder con alguno de los coches almacenados en el fichero (en cuyo caso deben actualizarse los kilómetros recorridos) o no corresponder con ninguno (en cuyo caso se debe insertar como nuevo registro en el fichero, manteniendo el orden).

18. Una farmacia dispone de un fichero de registros llamado MEDIC.DAT con información sobre los medicamentos que vende. En él se almacenan para cada medicamento: su identificador, el nombre del medicamento y el stock disponible en la farmacia de ese medicamento. Para ello se utiliza el siguiente tipo de datos:

```
tMedicam=registro
        id_medic:entero
        nombre:cadena
        stock:entero
freg
```

Además, la farmacia crea diariamente un fichero de registros llamado VENTAS.DAT con información sobre las ventas de medicamentos realizadas ese día. En él se almacena, para cada venta realizada, el identificador del medicamento vendido y la cantidad vendida (en una venta de un medicamento se pueden vender más de una unidad del mismo). Se utiliza el siguiente tipo de datos:

```
tVenta=registro
       id_medic: entero
       unidades:entero
freg
```

El fichero VENTAS.DAT está ordenado crecientemente por identificador de medicamento.

Escribir un subprograma que a partir de un fichero físico con información de los medicamentos de la farmacia y de un fichero físico que recoge las ventas realizadas en un día, actualice el fichero de medicamentos (en concreto el campo *stock*) a partir del de ventas.