

PRÁCTICA DE ARCHIVOS SECUENCIALES PARTE 1

Ejercicio 1

Un gimnasio tiene un archivo de inscripción en las cuatro disciplinas que realiza (**INSCCRIPCION**), secuencial, ordenado por DNI ascendente, que contiene un registro por cada asistente inscripto en una disciplina deportiva.

El diseño de registro de INSCPCION es: DNI, Apellido y Nombre; Disciplina ("F", "N", "T", "G" que representan fútbol, natación, taekwondo, gimnasia).

Se pide:

1. Generar cuatro archivos secuenciales, correspondientes a la inscripción de cada una de las disciplinas (**FUTBOL, NATACION, TAEKONDO, GIMNASIA**). El diseño de registro de los mismos es idéntico al del archivo de entrada.
2. Al finalizar el proceso, emitir la cantidad de inscriptos en cada disciplina.
3. Modificar el algoritmos propuesto teniendo en cuenta que los archivos de salida deben tener solamente los campos DNI y apellido
4. Modificar el algoritmo teniendo en cuenta que el campo disciplina puede estar con error. En este caso se debe grabar un archivo de errores.

Ejercicio 2

Un tradicional colegio secundario necesita procesar el resultado de los exámenes de ingreso rendido por numerosos aspirantes.

Para ello cuenta con un archivo **INGRESO**, secuencial, ordenado por DNI del aspirante ascendente,. Cada registro contiene el puntaje obtenido en c/u de las 3 evaluaciones de las 4 materias que se evalúan y un campo de puntaje por puntualidad que tendrá 8, si el aspirante asistió puntualmente a los 12 exámenes y 0 en caso contrario.

Se debe generar el archivo **APROBADOS**, que contendrá un registro por cada aspirante (con el DNI y la suma de los puntajes) que cumpla las siguientes condiciones:

1. Tener un puntaje mínimo de 4 en todas las evaluaciones, y
2. Tener un promedio mínimo de 6 en cada materia.

Sugerencia: utilizar el siguiente struct para el archivo INGRESO

```
struct tRIngreso {  
    unsigned int DNI ; // entero positivo  
    unsigned int asistencia ; // debe tener un valor 0 (si faltó a algún examen, u 8 si asistió a todos  
    int exámenes[4][3] ; // matriz de 4 filas = materias; 3 columnas=exámenes  
};
```

Ejercicio 3

Una entidad de medicina prepaga, desea recategorizar los planes de atención médica de sus asociados en base a la edad de cada uno de ellos. A su vez, los asociados están distribuidos en distintas zonas geográficas. Contamos con el archivo de **asociados** (ya generado) que se detalla a continuación:

Archivo **ASOCIADOS**: secuencial, está ordenado en forma ascendente por **Zona - Plan actual**. Contiene un registro por asociado con: Zona (1 carácter; ID plan actual (5 caracteres); Edad; DNI asociado

También tenemos un vector donde se detallan las edades **desde-hasta** válidas para cada uno de los 8 planes de atención médica:, donde cada celda tiene los siguientes campos: ID plan; edad desde; edad hasta

Ejemplo `tVPlanes vPlanes = { {"PLAN1", 0, 10 }, {"PLAN2", 11, 15}, {"PLAN3", 16, 20}, {"PLAN4", 21, 30}, {"PLAN5", 31, 40}, {"PLAN6", 41, 50}, {"PLAN7", 51, 60}, {"PLAN8", 61, 99} };`

Se debe generar un archivo **ACTUALIZADO**: secuencial, ordenado en forma ascendente por. Se indicará el nuevo plan para el asociado según su edad y los planes existentes. Contiene un registro por asociado con: Zona (1 carácter); ID plan actual (5 caracteres); ID nuevo plan (5 caracteres; DNI asociado

Ejercicio 4

Una escuela de buceo cuenta con un archivo secuencial (ordenado por código de elemento) en el cual se registran los materiales existentes: ropa, aletas, snorkel, lunetas, cilindros, reguladores, etc.

Los campos del registro del archivo son: código elemento (entero de 1 a 10); descripción; cantidad

Una escuela de buceo cuenta con un archivo secuencial (ordenado por código de elemento y con factibilidad 0/1) en el cual se registran los materiales existentes: ropa, aletas, snorkel, lunetas, cilindros, reguladores, etc.

Desarrollar un algoritmo que genere un nuevo archivo, a partir de los archivos preexistentes, en un proceso de intercalación, ordenado de igual forma que los archivos originales.

Pensar cómo generalizar el algoritmo para intercalar los archivos de cinco sucursales.

SUGERENCIA: PENSAR EN UN VECTOR DE REGISTROS . SE TRATA DE UN VECTOR DE 5 CELDAS, DONDE CADA CELDA COINCIDE CON CADA UNO DE LOS ARCHIVOS DE SUCURSALES. ASÍ LA LECTURA DE LA SUCURSAL 1 SE HACE SOBRE LA PRIMERA CELDA Y ASÍ SUCESIVAMENTE. FINALMENTE SE BUSCA EN EL VECTOR LA CLAVE MENOR QUE SE CORRESPONDE CON EL REGISTRO DEL ARCHIVO DE SUCURSAL QUE SE DEBE ATENDER.

Ejercicio 5

Para realizar el escrutinio de una elección se dispone de un archivo secuencial que contiene todos los **VOTOS** ordenado en forma ascendente por código de zona, código de distrito y mesa, siendo la estructura de su registro la que se detalla a continuación: código zona (entero 1 a 70); código distrito (entero 1 a 40); código de mesa (entero 1 a 50); número de lista votada (entero 1 a 8). Se trata de todos los votos de las 70 zonas, los 40 distritos de cada zona y las 50 mesas de cada zona/distrito.

Se pide desarrollar un algoritmo que genere el listado con los totales de votos por mesa, distrito, zona y mesa

SUGERENCIA: PENSAR EN UN ALGORITMO DEL TIPO CORTE DE CONTROL (LA INFORMACIÓN ESTÁ ORDENADA POR ZONA DISTRITO)

Zona:

Distrito:

Mesa Lista Ganadora

...

...

Listas sin votos en el distrito: ..., ..., ...

Distrito:

Mesa Lista Ganadora

...

Listas sin votos en el distrito: ..., ..., ...

Lista ganadora Zona: : Votos:

Zona:

Distrito:

Mesa Lista Ganadora

...

Listas sin votos en el distrito: ..., ..., ...

Lista ganadora Zona: : Votos:

¿Cómo resuelve este problema si la información está desordenada?