

Ejercicios: Archivos de Disco

Por favor utilice nombres de variables significativos en sus resoluciones, tratando de evitar CONT, X, Y, FLAG, etc... Describa en el nombre de la variable lo que ésta contiene: CONT_VENTAS, ACUM_SUELDO, PUNT_MAYOR, etc... De lo contrario el programa es muy difícil de leer y comprender.

Nota: muchos ejercicios se pueden resolver haciendo uso de memoria auxiliar gestionada de manera dinámica (malloc). Asegúrese de poder resolver con y sin esta herramienta de ser posible.

1) Realizar un programa que busque una palabra en un archivo de texto. Indique la cantidad de ocurrencias de la palabra, o bien su ausencia. El usuario ingresará el nombre del archivo en donde buscar y la palabra que será buscada. La palabra puede tener un máximo de 10 caracteres.

2) Realizar un programa que maneje un archivo de longitud desconocida, donde cada registro contiene el nombre de un alumno y cuatro notas. Hacerlo con un menú que permita crear el archivo, calcular el promedio, mostrarlo, buscar un registro determinado, modificar una nota, agregar registros y ordenar el archivo alfabéticamente.

3) Se tiene un archivo lista.dat que contiene la base de datos de artículos de un negocio: número de artículo (int), descripción (string de 30 caracteres), precio (float), proveedor (string de 30) y stock (int). Actualizar esta base de datos aplicando un 20% de aumento a todos los artículos del proveedor "PEREZ". Para probar el funcionamiento deberá crear primero el archivo con un programa auxiliar.

4) Se dispone del mismo archivo lista.dat del ejercicio anterior y de un archivo adicional llamado novedades.dat que contiene los nombres de proveedores (string de 20) y porcentaje de aumentos de precio (int). Actualizar lista.dat según lo que indica novedades.dat.

5) Dado el mismo archivo con artículos lista.dat, mostrar en pantalla y almacenar en un archivo la "nota de pedido" para completar el stock mínimo de artículos. El stock mínimo de cada artículo está almacenado en un archivo llamado SM.dat estructurado como n° de artículo (int) y cantidad mínima (int). Se pide armar una lista de productos que son necesarios reponer y cuántos deben comprarse de cada uno (como mínimo) para alcanzar

el nivel de stock mínimo. El archivo con la nota de pedido debe poder abrirse con un editor de texto normal.

6) Hacer un programa que ordene un archivo binario que contiene los datos de los alumnos de un curso (nombre, edad, sexo) en forma alfabética aplicando burbujeo simple, directamente sobre el archivo.

7) Dado el archivo ya ordenado del ejercicio anterior, ingresar un nombre por teclado, buscarlo en el archivo utilizando un algoritmo de búsqueda binaria y mostrar en pantalla los datos que le correspondan si es encontrado, o bien un mensaje de error en su ausencia.

8) Dado el archivo con artículos del punto 3, cree dos índices que permitan ver los artículos ordenados alfabéticamente por descripción, y el otro ver los artículos con stock menor a 5.

9) Realice un programa que lea un archivo que contiene un programa en lenguaje C y lo escriba en otro archivo pero omitiendo los saltos de línea. Esta operación busca optimizar el espacio en disco, partiendo del conocimiento de que para el compilador estos saltos no son significativos. Verifique que el programa “minimizado” se puede compilar y ejecutar normalmente del mismo modo que el original y que tiene menor tamaño.

10) Dentro de los archivos de audio MP3 se encuentra metainformación como ser el intérprete, nombre del álbum, género... Realice un programa que muestre esta información en pantalla.

La información se encuentra en los últimos 128 bytes del MP3, y se encuentra estructurada según esta definición:

```
struct datosMP3
{
    char tag[3]; //ojo, no es un string terminado en 0
    char titulo[30];
    char artista[30];
    char album[30];
    char anio[4]; //ojo, no es un string terminado en 0
    char comentario[30];
    unsigned char genero;
};
```

11) Se dispone de dos archivos binarios (ventas realizadas y provincias) compuestos respectivamente de estas estructuras:

```
struct venta {  
    char vendedor[60];  
    float monto;  
    int codigoprovincia;  
}  
  
struct provincia {  
    char nombre[40];  
    int codigoprovincia;  
}
```

Realice un programa que muestre en pantalla cuántas ventas se hicieron en cada provincia. Debe imprimir el nombre de las provincias y la cantidad de ventas. Los archivos originales solo están accesibles para modo lectura.

12) Dados los archivos del punto anterior, realice un programa que reciba por línea de comandos un código de provincia y cree un archivo de texto con los montos, uno debajo del otro, de todas las ventas realizadas en esa provincia. Guardarlos con dos decimales. Verificar que el archivo se puede abrir con un editor de texto regular.

13) Se dispone de un archivo binario con datos de artículos de una pinturería. Los artículos son de distintos proveedores. Realice un programa que cree un archivo de texto con los nombres de los proveedores sin repetir, uno debajo del otro.