



Gestión de Archivos en C

Prácticas de Laboratorio

- 1) Realizar un programa que contenga el siguiente menú de opciones

MENU PRINCIPAL

1. Agregar alumnos al archivo
2. Mostrar alumnos del archivo
3. Salir

El programa debe funcionar mediante la implementación de funciones. Agregar(), Mostrar(). Cada vez que se elige una opción se debe llamar a la función correspondiente, excepto cuando se presiona el 3, que el programa debe finalizar. Usar como nombre del archivo: "Alumnos.txt". Los datos a guardar en el archivo, son únicamente los "Nombres" de los alumnos del curso.

- 2) Hacer un programa que permita armar una estructura Proveedor con los siguientes datos:

- a. Nombre: Ej: Microsoft
- b. CUIT: 309765329
- c. Condición: Ej: activo

Ingresa 20 registros y Almacenarlos en un archivo binario.

El programa debe pedir que número de registro buscar. Ej: 5. El programa deberá mostrar el proveedor ubicado en la posición ingresada por teclado.

NOTA Utilizar fseek() para ir a la posición deseada.

- 3) Hacer un programa que guarde en un archivo binario 500 números aleatorios entre 0 y 1000. Al finalizar, se debe indicar el tamaño total del archivo en bytes.

NOTA: Utilizar las funciones fseek() y ftell() para resolver el problema. Se debe comprobar que el resultado midiendo el tamaño del archivo desde el sistema operativo



- 4) Se desea armar un archivo que contenga los productos de un supermercado. Dichos productos deberán tener los siguientes datos:

Tipo: Ej yerba
Marca: ej: cruz del sur
Tam: ej 1KG
Precio: Ej \$45
Stock: 100

Los datos de cada producto deben ser ingresados manualmente por teclado. Products.txt debe ser el nombre del archivo, y el mismo debe ser binario.

El programa deberá tener un menú de opciones muy similar al ejercicio anterior

MENU PRINCIPAL

1. Agregar productos
2. Listar productos
3. Salir

- 5) Modificar el programa anterior para agregar una nueva función que permita buscar en el archivo por producto y que muestre todos los datos asociados al producto.
- 6) Hacer un programa que permita guardar en un archivo binario (autos.bin), una estructura llamada AUTOS con los siguiente campos:

Marca: EJ: Ford
Modelo: EJ: Fiesta
Dominio: EJ: CEM321
Precio: EJ: 120000

El programa deberá tener un menú de opciones para:

- 1) Añadir nuevos autos al final
- 2) Listar todos los registros
- 3) Buscar x marca o modelo
- 4) Vaciar archivo
- 5) Salir

- ✓ Cada opción debe llamar a una función.
- ✓ Añadir: Implica agregar nuevos autos. Se deben ingresar por teclado y guardar en el archivo autos.txt
- ✓ Listar: implica contar la cantidad de registros que existen. Leerlos desde el archivo y mostrarlos en pantalla.



- ✓ Buscar x marca o modelo: implica ingresar una cadena de texto a buscar. Implementar función strstr () de C. el resultado es buscar un texto por ej: BMW (y que muestre en pantalla todo lo que se encuentra en el archivo con esa frase).
- ✓ Vaciar Archivo: Implica borrar el contenido del archivo (OJO; no se pide borrar el archivo sino su contenido).
- ✓ Salir: Sale del programa. Mientras no se pulse otra opción, el menú debe aparecer siempre.

7) Hacer una agenda telefónica, utilizando funciones y archivos binarios que permita:

- a) Ingresar datos de amigos
- b) Listar a los amigos
- c) Buscar por nombre
- d) Informar la cantidad de amigos existentes
- e) Vaciar el archivo
- f) Eliminar el archivo
- g) Salir

8) Implementando funciones, y archivos binarios en C se pide:

Crear una estructura de datos ALUMNO y guardarla en un archivo binario. La estructura debe contener:

Nombre CHAR 30
Apellido CHAR 20
Edad int
Año int
Promedio Académico float

El programa debe tener un menú de opciones para:

- a) Cargar Alumnos
- b) Listar Alumnos con edad ≥ 20 años
- c) Listar todo
- d) Export
- e) Salir

Cargar alumnos: debe permitir agregar nuevos alumnos al archivo. Si el archivo no existe se debe crear. Los datos se colocaran al final.



Listar Alumnos con edad ≥ 20 años: se debe recorrer el archivo y mostrar en pantalla todos los alumnos que tengan edad ≥ 20 años, mostrando únicamente el Nombre Apellido y edad

Listar todo: se debe mostrar en pantalla todo el contenido del archivo

Export: es una función que permite buscar todas las notas superiores a 9. Deberá tomar el nombre, apellido y nota de cada alumno que cumpla esa condición y guardar esos datos en otro archivo llamado "export.txt".

Salir: termina el programa. Hasta que no se seleccione esta opción, el programa no deberá terminar (luego de cada opción muestra nuevamente el menú)

- 9) Hacer un programa que permita crear una estructura de productos en MEMORIA DINAMICA, y que nos permita almacenar la informacion en un archivo binario. Visualizar el archivo creado. Utilizar funciones para todo.

- 10) Hacer un programa que permita cargar datos en una estructura empleado, la cual debe contener los siguientes miembros:

```
long ID_Empleado
char nombre[20]
char apellido[20]
float sueldo
int status // 0 inactivo, 1 empleado activo
```

*utilice memoria dinámica en lugar de la memoria estática para almacenar todos los registros que se quieran de empleados. El programa deberá tener un menú de opciones.

- 1) Cargar nuevo empleados // carga datos en la estructura y los almacena en archivo binario
- 2) buscar empleados inactivos // busca empleados inactivos y los muestra en pantalla
- 3) exportar datos
- 4) Salir

Exportar datos: Almacena en un archivo .TXT los datos de los empleados con estado 0 (Inactivo) Los datos de los empleados (cargados en la estructura) deben estar en un archivo binario llamado empleados.bin

PD: recuerde realizar las comprobaciones de asignación de memoria y de apertura de archivos, en todos los casos que corresponda, así como también, implementar funciones de fclose() y free() cuando sea necesario.