

TP FINAL

Consideraciones: El trabajo es individual. La entrega estará vigente desde el 14/6/2021 hasta el 25/6/2021, fecha en la cual cierra el cuatrimestre. Se entrega en una carpeta con nombre y apellido:

- .el código fuente,
- .el archivo de datos original cargado con al menos 10 registros y el
- .nuevo archivo obtenido con la baja física,
- .todo en carpeta comprimida.

Para la entrega estará disponible una carpeta en el campus, denominada "TP Final".

Datos:

Una empresa necesita un programa que almacene en un archivo binario N datos de sus clientes, según la siguiente estructura:

- ✓ número de id (ej: 8),
- ✓ apellido (ej: GARCIA)
- ✓ nombre (ej: PEDRO)
- ✓ categoría (ej: A, B ó C)
- ✓ facturación (ej: 65236,50).
- ✓ activo: (ej: 1, si está activo, 0 si fue dado de baja lógica)

Pre-condiciones:

El programa deberá:

1. Documentar los nombres de archivo que se utilizaron y su extensión.
2. El nombre de las variables y su función deben documentarse al principio del programa.
3. Utilizar librerías propias.
4. Tener en cuenta las funciones que se pueden reutilizar.
5. Validar todos los ingresos de datos y documentarlos.
6. Convertir a mayúsculas las cadenas o caracteres que se utilicen y así grabarlas en el archivo.
7. Controlar al menos 5 errores y documentarlos.
8. Tener los procesos y funciones debidamente documentados: pasaje de parámetros, por qué, que se espera de ese proceso función, etc.
9. El código debe respetar la indentación.
10. Tener nombre y apellido del autor y fecha de entrega.
11. Tener menú de opciones para operarlo.
12. **NO** se deben utilizar variables globales.

Requerimiento:

a. **Crear** el archivo clientes.dat

b. **Alta** de datos de clientes, para ello se debe tener en cuenta que el número de id (siempre es un número positivo) indica la posición en el archivo, (acceso directo) por ejemplo: id = 7, deberá insertarse en el lugar correcto, para ello es necesario utilizar fseek(). Los campos de las filas intermedias que no fueron cargados deben quedar en 0 (cero). Ej:

id	nom	ape	cat	fact	act
0	0	0	0	0	0
2	Lila	Perez	B	85432.7	1
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
0	0	0	0	0	0
7	Pepe	Luena	C	65236.5	1
0	0	0	0	0	0
9	Lucho	Aviles	B	45567.4	0

10	Mirna	Gauna	A	61456.7	1
----	-------	-------	---	---------	---

Si el número de id ingresado existe debe emitir un mensaje de error y pedir nuevamente el número de legajo.

- c. Listar los datos del archivo por pantalla, con tres opciones a elección del usuario:
 1. considerando dar a elegir al usuario listar todos (activos, pasivos y nulos) y
 2. considerando sólo el campo activo = 1. Los listados deben emitir sus respectivos títulos de columna. (acceso secuencial)
 3. considerando emitir el archivo resultado después de aplicar el punto i.
- d. Permitir buscar datos de un empleado con dos opciones:
 1. por id y
 2. por apellido

es decir que el usuario puede elegir cómo buscar. Emitir los datos y en caso que no se encuentre emitir el mensaje correspondiente (por id, acceso directo, por apellido, acceso secuencial).
- e. Permitir modificar de categoría e ingresos a un cliente, buscando por el número de id o por el apellido, a elección del usuario. En ambos casos el programa deberá preguntar:
 Está seguro que desea modificar los datos de...?. (por id acceso directo, por apellido, acceso secuencial)
- f. Permitir leer el archivo "clientes.dat" y actualizar la facturación de los clientes (podría ser debido a cambios o errores en la misma). Ej: se pide categoría, se ingresa la misma por teclado, digamos B y su respectivo incremento o decremento en porcentaje: por ejemplo 7. Esta función debe realizar los cálculos para la categoría ingresada y registrar la nueva facturación en el archivo. (acceso secuencial).
- g. Permitir emitir los datos ordenados por apellido (A-Z). Para ello se debe utilizar (sólo para este caso) un array de struct, donde se cargarán los datos del archivo para ordenarlos y emitirlos por pantalla. No se debe registrar el resultado de este proceso en un archivo
- h. Permitir dar de baja lógica a un cliente ingresando su número de legajo (acceso directo). La baja lógica significa modificar el campo activo, si está en 1 debe asignarse un 0, si está en 0, debe dar un mensaje de error y volver al menú. Antes de dar la baja, el programa debe emitir los datos y pedir confirmación. Ej: Está seguro que quiere dar de baja a ...? una vez hecha la baja se deberá emitir el plantel de clientes activos.
- i. Permitir dar la baja física (eliminar) datos de todos los clientes inactivos, para ello se debe construir un nuevo archivo. (acceso secuencial).

Post-condiciones:

1. El programa deberá, en todos los casos, emitir los resultados de forma clara, sin elementos que den lugar a confusión.
2. Los puntos no resueltos deben documentarse también.

Será valorada la correspondencia con la consigna, la codificación, la organización, la prolijidad, la claridad y la documentación de todo el desarrollo.