

Introducción a las Bases de Datos

Fundamentos de Organización de Datos

Práctica 3

Archivos Secuenciales - Bajas - Registros de longitud variable

1. Realizar un programa que genere un archivo de productos. De cada producto se registra: código, nombre, cantidad actual, cantidad mínima, cantidad máxima y precio. El programa debe presentar un menú con las siguientes opciones:
 - a. Crear el archivo y cargarlo a partir de datos ingresados por pantalla. En el primer registro del archivo a generar se debe guardar un registro cabecera para mantener la lista de espacios libres, inicializado con el valor cero (0) en el campo código de producto, el cual indica que no hay espacio libre.
 - b. Abrir un archivo existente y permitir su mantenimiento con control de espacio libre a partir de una lista encadenada de organización LIFO (Last In, First Out). La lista tendrá la cabecera en el primer registro del archivo. El encadenamiento de registros debe realizarse con el campo reservado para el código de producto, pero especificando los números de registro referenciados con signo negativo. Una vez abierto el archivo, brindar operaciones para:
 - i. Dar de alta un producto leyendo la información desde teclado. Para esta operación, en caso de ser posible, deberá recuperarse el espacio libre. Es decir, si en el campo correspondiente al código de producto del registro cabecera hay un valor negativo, por ejemplo -20, se debe leer el registro en la posición 20, copiarlo en la posición 0 (actualizar la lista de espacio libre) y grabar el nuevo registro en la posición 20. Con el valor 0 (cero) en el registro cabecera se indica que no hay espacio libre.
 - ii. Modificar los datos de un producto leyendo la información desde teclado. El código de producto no puede ser modificado.
 - iii. Eliminar un producto cuyo código es ingresado por teclado. Por ejemplo, si se da de baja un registro en la posición 20, en el campo código de producto

del registro cabecera deberá figurar -20, y en el registro en la posición 20 debe copiarse el antiguo registro cabecera.

- c. Listar en un archivo de texto todos los productos, incluyendo los borrados, que representan la lista de espacio libre. El archivo debe llamarse "reporte_productos.txt".

NOTA: Tanto en la creación como en la apertura el nombre del archivo debe ser proporcionado por el usuario.

2. Modificar el ejercicio 5 de la práctica 1 (programa de gestión de datos de personas) , agregándole una opción para realizar bajas copiando el último registro del archivo en la posición del registro a borrar y luego truncando el archivo en la posición del último registro de forma tal de evitar duplicados.
3. Definir un programa que genere un archivo con los datos personales de los docentes de la facultad, utilizando para ello registros de longitud fija. Implementar un algoritmo que, a partir del archivo de datos generado, elimine de forma **lógica** los docentes de más de 65 años.
4. Dada la siguiente estructura:

type

```
tNombre          = String[50];  
tArchPersonas = file of tNombre;
```

Las bajas se realizan apilando registros borrados y las altas reutilizando registros borrados. El registro 0 se usa como cabecera de la pila de registros borrados: el número 0 implica que no hay registros borrados y N indica que el próximo registro a reutilizar es el N, siendo éste un número relativo de registro válido.

- a. Implemente el siguiente módulo:

{Abre el archivo y agrega el nombre recibido como parámetro manteniendo la política descrita anteriormente}

```
procedure agregar (var a: tArchPersonas; nombre: string);
```

b. Liste el contenido del archivo omitiendo los registros eliminados. Modifique lo que considere necesario para obtener el listado.

5. Dada la siguiente estructura:

```
type  
  
tNombre          = String[50];  
  
tArchPersonas = file of tNombre;
```

Las bajas se realizan apilando registros borrados y las altas reutilizando registros borrados. El registro 0 se usa como cabecera de la pila de registros borrados: 0 implica que no hay registros borrados y N indica que el próximo registro a reutilizar es el N, siendo éste un número relativo de registro válido.

Implemente el siguiente modulo:

```
{Abre el archivo y elimina el nombre recibido como parámetro  
manteniendo la política descrita anteriormente}
```

```
procedure eliminar (var a: tArchPersonas; nombre: string);
```

6. Modificar el programa de gestión de medicamentos de la práctica 1 agregándole opciones para:

- a. Realizar la baja lógica de un medicamento dado su nombre y presentación. La baja lógica de un medicamento debe realizarse cambiándole el signo (multiplicando por -1) al número de medicamento.
- b. Listar en un archivo de texto todos los medicamentos, excepto aquellos que se encuentren borrados lógicamente.
- c. Efectivizar las bajas compactando el archivo. Para esto debe copiar el archivo sin los registros borrados lógicamente, eliminar el archivo viejo y renombrar el nuevo con el nombre original.

7. Realizar un programa con opciones para:

- a. Crear un archivo de nombres de personas.
- b. Abrir un archivo existente para listar su contenido en pantalla.

Para la creación, los nombres se ingresan desde teclado, y la carga finaliza con un nombre igual a nulo, el cual no se incorpora al archivo. Los registros se almacenan con longitud variable.

8. Hacer un programa con opciones para la gestión de datos de personas. De cada persona se debe registrar: apellido, nombre y fecha de nacimiento. El programa debe cumplir con las siguientes especificaciones:

- Los registros deben almacenarse con longitud variable (el apellido y el nombre se guardan con la longitud que tengan, con un byte prefijo para precisar la longitud del mismo) en un archivo sin tipo, y con un byte de control inicial para indicar la longitud total del registro.
- Las bajas deben ser lógicas, sustituyendo la primera letra del apellido por un asterisco.
- Se debe agregar una opción para compactar el archivo eliminando los registros marcados como eliminados: se copia el archivo secuencialmente ignorando los registros marcados, se borra el archivo original y se renombra el archivo nuevo con el nombre del original.
- El campo fecha debe guardarse en formato *LongInt*, dado que no cambia de tamaño al ser modificada (el registro se puede actualizar reescribiendo sólo ese campo, de 4 bytes).
- El archivo debe abrirse con tamaño de unidades de transferencia de 1 byte: `Reset (arch, 1)`. La lectura de cada registro debe efectuarse con dos lecturas de bloque (`BlockRead`): una para el tamaño del registro, en un byte, y otra para el registro en sí, en un arreglo de bytes sobre el que se trabajará en memoria para copiar los campos a variables (con el procedimiento `Move`).

9. Realizar un programa con opciones para:

- a. Crear un archivo con información de medicamentos de una farmacia.
- b. Abrir un archivo existente para ver su contenido en pantalla.

Para la creación, la información se ingresa desde teclado en el siguiente orden: nombre, descripción y, por último, stock. La carga finaliza con un nombre de medicamento igual a nulo, el cual no se incorpora al archivo. Los registros se almacenan con longitud variable. Se debe usar el carácter # como separador de registros y @ como separador de campo.