

Introducción a las Bases de Datos

Fundamentos de Organización de Datos

Práctica 3

Archivos Secuenciales - Bajas - Registros de longitud variable

1. Modificar el ejercicio 4 de la práctica 1 (programa de gestión 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.
2. Definir un programa que genere un archivo con registros de longitud fija conteniendo información de empleados de una empresa aseguradora. Se deberá almacenar: código de empleado, nombre y apellido, dirección, telefono, D.N.I y fecha nacimiento. Implementar un algoritmo que, a partir del archivo de datos generado, elimine de forma **lógica** todos los empleados con DNI inferior a 5.000.000.

Para ello se podrá utilizar algún carácter especial delante de algún campo String a su elección. Ejemplo: '*Juan'.

3. Realizar un programa que genere un archivo de novelas filmadas durante el presente año. De cada novela se registra: código, género, nombre, duración, director y precio. El programa debe presentar un menú con las siguientes opciones:
 - a. Crear el archivo y cargarlo a partir de datos ingresados por teclado. Se utiliza la técnica de lista invertida para recuperar espacio libre en el archivo. Para ello, durante la creación del archivo, en el primer registro del mismo se debe almacenar la cabecera de la lista. Es decir un registro ficticio, inicializando con el valor cero (0) el campo correspondiente al

código de novela, el cual indica que no hay espacio libre dentro del archivo.

- b. Abrir el archivo existente y permitir su mantenimiento teniendo en cuenta el inciso a., se utiliza lista invertida para recuperación de espacio. En particular, para el campo de 'enlace' de la lista, se debe especificar los números de registro referenciados con signo negativo, (utilice el código de novela como enlace). Una vez abierto el archivo, brindar operaciones para:
 - i. Dar de alta una novela 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 novela del registro cabecera hay un valor negativo, por ejemplo -5, se debe leer el registro en la posición 5, copiarlo en la posición 0 (actualizar la lista de espacio libre) y grabar el nuevo registro en la posición 5. Con el valor 0 (cero) en el registro cabecera se indica que no hay espacio libre.
 - ii. Modificar los datos de una novela leyendo la información desde teclado. El código de novela no puede ser modificado.
 - iii. Eliminar una novela cuyo código es ingresado por teclado. Por ejemplo, si se da de baja un registro en la posición 8, en el campo código de novela del registro cabecera deberá figurar -8, y en el registro en la posición 8 debe copiarse el antiguo registro cabecera.
- c. Listar en un archivo de texto todas las novelas, incluyendo las borradas, que representan la lista de espacio libre. El archivo debe llamarse "novelas.txt".

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

4. Dada la siguiente estructura:

type

```
tTitulo      = String[50];  
  
tArchLibros = file of tTitulo ;
```

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 título del libro, recibido como parámetro manteniendo la política descripta anteriormente}

procedure agregar (**var** a: tArchLibros ; titulo: string);

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

5. Dada la estructura planteada en el ejercicio anterior, implemente el siguiente módulo:

{Abre el archivo y elimina el título del libro recibido como parámetro manteniendo la política descripta anteriormente}

procedure eliminar (**var** a: tArchLibros ; titulo: string);

6. Una cadena de tiendas de indumentaria posee un archivo maestro no ordenado con la información correspondiente a las prendas que se encuentran a la venta. De cada prenda se registra: cod_prenda, descripción, colores, tipo_prenda, stock y precio_unitario. Ante un eventual cambio de temporada, se deben actualizar las prendas a la venta. Para ello reciben un archivo conteniendo: cod_prenda de las prendas que quedarán obsoletas. Deberá implementar un procedimiento que reciba ambos archivos y realice la baja lógica de las prendas, para ello deberá modificar el stock de la prenda correspondiente a valor negativo.

Por último, una vez finalizadas las bajas lógicas, deberá efectivizar las mismas compactando el archivo. Para ello no podrá utilizar ninguna estructura auxiliar, debe resolverlo dentro del mismo archivo. Solo deben quedar en el archivo las prendas que no fueron borradas, una vez realizadas todas las bajas físicas.

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. Los nombres deben incorporarse al archivo separandolos con el caracter '@'.

8. Hacer un programa con opciones para la gestión de datos de empleados. De cada empleado se debe registrar: código Empleado, DNI, apellido, nombre, peso, estatura y fecha de nacimiento. El programa debe cumplir con las siguientes especificaciones:

- Se almacena la información en un archivo sin tipo.
- Los registros deben almacenarse con longitud variable (el apellido, el nombre y el DNI, se guardan con la longitud que tengan, con bytes prefijos para precisar la longitud del campo correspondiente)
- y con bytes de control inicial para indicar la longitud total del registro.

Nota: el tamaño del registro y de los campos variables se almacena como integer por eso se necesita más de 1 byte

- El campo fecha debe guardarse en formato *LongInt*.
- El campo código empleado debe ser un integer, y los campos pesos y estatura de tipo real.
- 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 registros de longitud variable conteniendo información de productos de una empresa de tecnología. De cada producto es necesario almacenar: nombre, descripción, precio, stock mínimo y stock actual. Para la separación de campos debe utilizar el caracter "|" (pipe) y para la separación de registros debe utilizar el caracter "*" (asterisco). El nombre del archivo lo debe ingresar el usuario desde teclado y la carga finaliza cuando el usuario ingresa como nombre la palabra "fin".

- b. Abrir el archivo creado en el punto anterior e informar el contenido por la salida estándar (pantalla). El listado que debe presentar deberá contar con el siguiente formato para cada registro:

INFORMACIÓN DE REGISTRO

NOMBRE PRODUCTO: - - - - -

DESCRIPCIÓN PRODUCTO: - - - - -

STOCK MÍNIMO: - - - - -

STOCK ACTUAL: - - - - -