

## Algoritmica clasica

### Merge

metros construidos. **Nota:** todos los archivos estan ordenados por `cod_zona`.

9. Una federación de competidores de running organiza distintas carreras cada mes. Cada carrera cuenta con DNI de corredor, apellido, nombre, kilómetros que corrió, y si ganó o no la carrera (valor 1 indica que ganó, y valor 0 indica que no ganó la carrera)

Puede haber distintas cantidades de carreras en el mes. Para el mes de abril se organizaron 5 carreras.

Escriba el programa principal con la declaración de tipos necesaria y realice un proceso que reciba los 5 archivos y genere el archivo maestro con la siguiente información: DNI, apellido, nombre, kilómetros totales y cantidad de carreras ganadas.

Todos los archivos se encuentran ordenados por DNI del corredor. Cada persona puede haber corrido una o más carreras.

### Corte de Control

5. La Dirección de Población Vulnerable del Ministerio de Salud y Desarrollo Social solicita información a cada municipio indicando cantidad de niños y de adultos mayores que están en situación de riesgo debido a la situación socioeconómica del país. Para ello se recibe un archivo indicando: partido, localidad, barrio, cantidad de niños y cantidad de adultos mayores.  
Se sabe que el archivo se encuentra ordenado por partido y localidad. Se pide imprimir por pantalla con el siguiente formato:

**Partido:**

**Localidad 1:**

**Cantidad niños: \_\_\_\_\_ Cantidad adultos: \_\_\_\_\_**

**Total niños localidad 1:----- Total adultos localidad 1:-----**

**Localidad 2:**

**Cantidad niños: \_\_\_\_\_ Cantidad adultos: \_\_\_\_\_**

**Total niños localidad 2:----- Total adultos localidad 2:-----**

**TOTAL NIÑOS PARTIDO:----- TOTAL ADULTOS PARTIDO:-----**

### Actualizacion N detalles

## Archivos

Se desea actualizar un archivo maestro a partir de 500 archivos detalle de votos de localidades.

Cada archivo detalle contiene código de provincia, código de localidad, cantidad de votos válidos, cantidad de votos en blanco y cantidad de votos anulados. El archivo se encuentra ordenado por código de provincia y código de localidad.

El archivo maestro tiene código de provincia, nombre de provincia, cantidad total de votos válidos, cantidad total de votos en blanco y cantidad total de votos anulados. El archivo se encuentra ordenado por código de provincia.

Realizar la actualización de archivo maestro con la información de los archivos detalles. Además al final se debe informar en un archivo de texto denominado cantidad\_votos\_04\_07\_2023.txt la cantidad de archivos procesados, la cantidad total de votos válidos, cantidad total de votos en blanco y cantidad total de votos anulados de los archivos detalles con el siguiente formato:

Cantidad de archivos procesados : \_\_\_\_

Cantidad Total de votos: \_\_\_\_

Cantidad de votos válidos: \_\_\_\_

Cantidad de votos anulados: \_\_\_\_

Cantidad de votos en blanco: \_\_\_\_

Se debe realizar el programa completo con sus declaraciones de tipo.

*Resolucion Profe*

## Baja Logica

5. Se cuenta con un archivo de artículos deportivos a la venta. De cada artículo se almacena: nro de artículo, descripción, color, talla, stock disponible y precio del producto. Se reciben por teclado los nros de artículos a eliminar, ya que no se fabricarán más. Se deberá realizar la baja lógica de los artículos correspondientes. Además, se requiere listar en un archivo de texto todos los artículos

---

Página 2 de 3

---

Archivos Bajas

eliminados, para ello debe almacenar toda la información del artículo eliminado en el archivo de texto. (**No debe recorrer nuevamente** el archivo maestro, deberá hacerlo en simultáneo). Escriba el programa principal con la declaración de tipos necesaria y realice un proceso que reciba el archivo maestro y actualice el archivo maestro a partir de los códigos de artículos a borrar. El archivo maestro se encuentra **ordenado** por el código de artículo.

## Baja Fisica

4. Una disquera cuenta con un archivo conteniendo la información de discos(cd) que posee a la venta. De cada cd se conoce: un código único, nombre álbum, género, artista una descripción asociada, año de edición y cantidad de copias en stock. Al archivo no tiene orden.

Trimestralmente la disquera actualiza el archivo modificando los discos de los que ya no posee stock. Implementar un procedimiento que modifique el stock a 0 de los discos obsoletos e informe por pantalla nombre de álbumes que quedaron sin stock. Se deberá además declarar los tipos de datos necesarios y la llamada al procedimiento de modificación. Para ello el usuario ingresará por teclado los códigos de cd que ya no tienen stock.

Además, se deberá implementar la compactación del archivo, es decir un procedimiento que reciba el archivo de discos y elimine físicamente los discos que no tienen stock.

## Lista Invertida

2. Se dispone de un archivo que contiene información de autos en alquiler de una rentadora. Se sabe que el archivo utiliza la técnica de lista invertida para aprovechamiento de espacio. Es decir las bajas se realizan apilando registros borrados y las altas reutilizando registros borrados. El registro en la posición 0 del archivo se usa como cabecera de la pila de registros borrados.

```
Type  
  
tVehiculo= Record  
    codigoVehiculo:integer;  
    patente: String;  
    #motor:String;  
    cantidadPuertas: integer;  
    precio:real;  
    descripcion:String  
end;  
tArchivo = File of tVehiculo;
```

**Nota:** El valor '0' en el campo descripción significa que no existen registros borrados, y 'N' indica que el próximo registro a reutilizar es el N, siendo éste un número relativo de registro válido.

Se solicita implementar los siguientes módulos:

*{Abre el archivo y agrega un vehículo para alquiler, el mismo se recibe como parámetro y debe utilizar la política descrita anteriormente para recuperación de espacio}*

```
Procedure agregar (var arch: tArchivo; vehiculo: tVehiculo);  
var
```

---

Página 1 de 3

---

Archivos Bajas

```
Begin  
End;
```

*{Abre el archivo y elimina el vehículo que posea el código recibido como parámetro manteniendo la política descrita anteriormente}*

```
Procedure eliminar (var arch: tArchivo; codigoVehiculo: integer);  
var  
Begin  
End;
```

## Arboles

### B

- 4) Dado el árbol B que se detalla más abajo, con orden 6, es decir, capacidad de 5 claves como máximo. Muestre los estados sucesivos al realizar la siguiente secuencia de operaciones: +425, -60 y -22 **Utilice política izquierda y derecha para resolver underflow.**

Nodo 0: 5, i, 2(60)1(90)3(107)4(287)5(400)6  
 Nodo 2: 2, h, (12)(22)  
 Nodo 1: 2, h, (63)(66)  
 Nodo 3: 2, h, (94)(95)  
 Nodo 4: 3, h, (116)(129)(280)  
 Nodo 5: 2, h, (328)(358)  
 Nodo 6: 5, h, (403)(404)(506)(518)(620)

- 5) Dado el árbol B que se detalla a continuación, con orden 6, es decir, capacidad de 5 claves como máximo. Muestre los estados sucesivos al realizar la siguiente secuencia de operaciones: alta 756, baja 300 y baja 85. **Indique L/E de nodos. Dibuje el árbol resultante de cada operación. Política de resolución de underflow derecha**

2

## Conceptos de Bases de Datos 2024

NODO 0: (i, 3 elem.), 1(85) 2(379) 3(755) 4  
 NODO 1: (h, 2 elem.), (14)(80)  
 NODO 2: (h, 2 elem.), (216)(300)  
 NODO 3: (h, 3 elem.), (601)(682)(695)  
 NODO 4: (h, 5 elem.), (771)(810)(853)(907)(964)

**B+**

- 6) Dado el árbol B+ que se detalla a continuación, con orden 5, es decir, capacidad de 4 claves como máximo. Muestre los estados sucesivos al realizar la siguiente secuencia de operaciones: alta de la clave 19, alta de 25 y finalmente baja de las claves 63 y 62. **Utilizando política de resolución de underflow izquierda y derecha. Dibuje el árbol resultante de cada operación. Indique lecturas y escrituras en cada operación.**

NODO 0 (i, 4 elemento/s) 2(23)1(54)3(62)4(79)5

NODO 2 (h, 4 elemento/s) (18) (20) (21) (22) Siguiente hoja: 1

NODO 1 (h, 4 elemento/s) (23) (26) (33) (38) Siguiente hoja: 3

NODO 3 (h, 1 elemento/s) (54) Siguiente hoja: 4

NODO 4 (h, 2 elemento/s) (62) (63) Siguiente hoja: 5

NODO 5 (h, 3 elemento/s) (79)(80) (91) Siguiente hoja: -1

- 7) Dado el árbol B +que se detalla siguiendo, con orden 7, es decir, capacidad de 6 claves como máximo. Muestre los estados sucesivos al realizar la siguiente secuencia de operaciones: -173, +172 y -92. **Utilice política derecha**

0: 2(165) 9

2: 10(92) 1(104) 3(130)4

9: 5(178) 6(264) 7(378) 8

10: (59)( 85)( 89) 1

5: (166)(167)(168)(169)(171)(173)6

1: (92)(94) 3

6: (178)(187)(197)(198) 7

3: (104)(120) 4

7: (264)(266)(269) 8

4: (130)(163) 5

8: (378)(390)(395)(400) -1

## Dispersion

### Progresiva

1. Dado el siguiente archivo dispersado más abajo, dibuje los estados sucesivos para las siguientes operaciones: +44, +65, +30, +66. Función de dispersión: **Clave MOD 11**. Al finalizar calcule la densidad de empaquetamiento.

Dirección	Registro	Registro
0	33	22
1	23	45
2		
3	36	
4	48	
5		
6		
7	51	
8		
9		
10	76	54

### Encadenada

4. Dado el siguiente archivo dispersado, dibuje los estados sucesivos para las siguientes operaciones: +57, +59, +64, -43, -39. Función de dispersión: **Clave MOD 7**. Técnica de resolución de colisiones: **Saturación progresiva encadenada**. Calcule la **densidad de empaquetamiento**. Justifique brevemente cada operación.

Dirección	Enlace	Registro
0	-1	
1	-1	43
2	-1	23
3	-1	
4	-1	39
5	-1	
6	-1	

**Area separada**

7. Dado el siguiente archivo dispersado, dibuje los estados sucesivos para las siguientes operaciones: +36, +62, -8, -41. Función de dispersión: **Clave MOD 7**. Técnica de resolución de colisiones: **Saturación progresiva encadenada con área de**

**desborde separada.** Justifique brevemente cada operación.

Dirección	Enlace	Registro	Registro
0	-1	14	
1	-1	8	22
2	-1		
3	-1	24	
4	1	11	18
5	-1		
6	2	20	27

Dirección	Enlace	Registro
0	-1	34
1	-1	60
2	0	41
3	-1	
4	-1	

**Dispersion doble**



8. Dado el siguiente archivo dispersado más abajo, dibuje los estados sucesivos para las siguientes operaciones: +68, +34, +24, -59, -51. Función de dispersión: *Clave MOD 11*. Segunda función de dispersión: *Clave MOD 6*. Al finalizar calcule la densidad de empaquetamiento.

Dirección	Registro
0	
1	23
2	
3	47
4	59
5	
6	50
7	51
8	30
9	
10	43

## Extensible

3. Para las claves siguientes, realice el proceso de dispersión mediante el método de hashing extensible, sabiendo que cada nodo tiene capacidad para dos registros. El número natural indica el orden de llegada de las claves. **Justifique brevemente**

1	Avena	00000001	2	Cebada	01100011
3	Trigo	11110101	4	Centeno	01010110
5	Arroz	00101000	6	Maíz	00110100
7	Sorgo	10110010	8	Quinoa	01111110
9	Kamut	01111100	10	Mijo	00110000