

## Redictado FOD 2016 – Tercera Fecha de Evaluación Parcial – 20/12/16

1. Una editorial de diarios y revistas distribuye y vende sus productos (diarios y revistas) al interior del país. La editorial cuenta con 5 distribuidoras. Cada distribuidora envía un listado con los productos vendidos indicando: código de producto y cantidad vendida del mismo.

La editorial posee un archivo maestro en donde almacena la información de todos los diarios y revistas que distribuye, para ello el archivo maestro cuenta con el código de producto, nombre, descripción y el stock actual de cada producto.

Escriba el programa principal con la declaración de los tipos de datos necesarios y realice un proceso que reciba los 5 detalles y actualice el archivo maestro con la información proveniente de los archivos de detalle.

Tanto el maestro como los detalles se encuentran ordenados por el código de producto. No se realizan ventas de ejemplares sin stock.

Nota: la solución planteada debe ser escalable en cuanto a la cantidad de detalles.

2. Dado el estado inicial del siguiente árbol B+, con nodos con capacidad máxima para tres registros en todos los nodos, y mínima 1. Especifique los estados sucesivos completos al realizarle las operaciones que se detallan a continuación. Además, indique los nodos que se leen y escriben para realizar cada operación, en el orden correspondiente.

Nota: Los nodos internos se parten 2-1, y los *underflows* se resuelven por defecto con hermanos derechos.

2 0 (348) 4 (747) 1 (812) 3			
0 60, 187, 223   4	4 348, 491, 512   1	1 747   3	3 933   -1

Operaciones: +300, -933, +400, -747

3. Dado el siguiente estado inicial de un archivo directo estático de bloques con capacidad para dos registros. Utilizando la función de dispersión  $h(x) = x \bmod 7$ , especifique los estados sucesivos completos al realizarle las operaciones detalladas a continuación, indicando los bloques que se leen y escriben para realizar cada una de ellas en el orden correspondiente, así como la densidad de empaquetamiento de cada estado.

Los registros en saturación se organizan con dispersión doble, con la función  $s(x) = x \bmod 3 + 1$ .

**Para las altas, se debe verificar que no haya claves duplicadas, y para las bajas, se sustituye la clave del registro a eliminar por -1.**

0: 322, 504

1: 162

2: 359, 562

3: 241, 598

4: 480

5: 281, -1

6: 531, 650

Operaciones: +510, -562, +301, +442.

## Pautas de Corrección

Los errores pueden ser inadmisibles, graves, moderados o leves.

Opciones para aprobar un tema: no tener errores inadmisibles / 1 grave y hasta 2 leves / 4 leves / 2 moderados y hasta 1 leve / 1 moderado y hasta 3 leves. Cualquier otra combinación desaprueba el tema.

### Archivos

**const**

CD = 5; *// cantidad de distribuidoras*

NoCod = 999999999;

**type**

tProd = **record** *// registro de producto*  
    cod: **longword**;  
    no, desc: **string**;  
    stock: **longword**  
    **end**;

tVta = **record** *// registro de ventas*  
    cod: **longword**;  
    cant: **word**  
    **end**;

tMae = **File of** tProd; *// archivo maestro de productos*

tDet = **File of** tVta; *// archivo detalle de ventas de productos*

tCtlDet = **record** *// registro de control de archivo de detalle*  
    det: tDet;  
    rDet: tVta  
    **end**;

tDetalles = **array**[1..CD] **of** tCtlDet;

**procedure** leerDet( **var** det: tDet; **var** rDet: tVta );

**begin**

**if not** eof(det)

**then** read(det, rDet) **else** rDet.cod:=NoCod

**end**;

**procedure** minimo( **var** ctlDet: tDetalles; **var** min: tVta );

**var** i, imin: **byte**;

**begin**

imin:=1;

**for** i:=2 **to** CD **do if** ctlDet[i].rDet.cod < ctlDet[imin].rDet.cod **then** imin:=i;

min:=ctlDet[imin].rDet;

**if** min.cod <> NoCod **then** leerDet(ctlDet[imin].det, ctlDet[imin].rDet)

**end**;

```

procedure actualizar( var ctlDet: tDetalles );
// Recibe los archivos de detalle asignados y sin abrir
var
    mae: tMae; // archivo maestro
    prod: tProd;
    codAct: longword;
    i: byte;
    min: tVta;

begin
assign(mae, 'productos');
reset(mae);
read(mae, prod);
for i:=1 to CD do begin reset(ctlDet[i].det); leerDet(ctlDet[i].det, ctlDet[i].rDet) end;
minimo(ctlDet, min);
while min.cod <> NoCod do begin
    codAct:=min.cod;
    while prod.cod<>codAct do read(mae, prod); // búsqueda del registro de producto
    while min.cod=codAct do begin
        prod.stock:=prod.stock-min.cant;
        minimo(ctlDet, min)
    end;
    seek(mae, filepos(mae)-1);
    write(mae, prod) // escritura del registro de producto actualizado
end;
close(mae);
for i:=1 to CD do close(ctlDet[i].det)
end;

var
    nomArch: string;
    detalle: tDetalles;
    i: byte;

begin
for i:=1 to CD do begin
    write('Ingrese nombre de archivo de detalle ', i, ': ');
    readln(nomArch);
    assign(detalle[i].det, nomArch)
end;
actualizar(detalle)
end.

```

## Errores posibles

No asignar y abrir con reset los 5 detalles: inadmisible.

No abrir correctamente (con *reset*) el archivo maestro: grave.

Corte de control principal por eof o valor alto del maestro (procesa registros sin sentido): si la solución es eficaz, moderado, si no, inadmisible.

No realizar un proceso que devuelva mínimo de los detalles: grave.

No inicializar por cada código diferente un contador de revistas: inadmisible.

Volver atrás en detalles: inadmisible.

No buscar en el maestro el registro a actualizar: inadmisible.

No volver atrás en el maestro para escribir: inadmisible.

No actualizar maestro: inadmisible.

No cerrar el maestro: grave.

No procesar en simultáneo detalles y maestro: inadmisible.

Recorrer más de una vez los archivos: inadmisible.

No hacer un procedimiento: leve.

Leer más de una vez al sacar el mínimo (perder registros): inadmisible (solo se avanza en el archivo que tenga el código mínimo).

No utilizar estructura auxiliar para trabajar los 5 detalles y los registros correspondientes a cada detalle: inadmisible, contradice el enunciado, la solución no es escalable.

## Árboles

2 0 (348) 4 (747) 1 (812) 3			
0 60, 187, 223   4	4 348, 491, 512   1	1 747   3	3 933   -1

+300: [L2] L0 E0 E5 E2 E6 E7

7 2 (747) 6				
2 0 (223) 5 (348) 4			6 1 (812) 3	
0 60, 187   5	5 223, 300   4	4 348, 491, 512   1	1 747   3	3 933   -1

-933: [L7] L6 L3 L1 E1 L2 E2 E6 E7

Libres: 3

7 2 (348) 6			
2 0 (223) 5		6 4 (747) 1	
0 60, 187   5	5 223, 300   4	4 348, 491, 512   1	1 747   -1

+400: [L7] L6 L4 E4 E3 E6

7 2 (348) 6				
2 0 (223) 5		6 4 (491) 3 (747) 1		
0 60, 187   5	5 223, 300   4	4 348, 400   3	3 491, 512   1	1 747   -1

-747: [L7] L6 L3 L1 E3 E1 E6

7 2 (384) 6				
2 0 (223) 5		6 4 (491) 3 (512) 1		
0 60, 187   5	5 223, 300   4	4 348, 400   3	3 491   1	1 512   -1

## Errores posibles

Renumerar nodos: inadmisible.

Olvidar enlaces en las hojas: inadmisible.

Agregar nodos nuevos en orden incorrecto alta 300: inadmisible (no se entiende el algoritmo). Por el recorrido de los nodos primero se agrega el 5 como hoja, luego el 6 como interno y finalmente el 7 como nueva raíz. Si se aclara la presunción de mantener la raíz en una posición fija, sería aceptable que la raíz permaneciera fija en la posición 2.

Omitir la lectura o escritura de un nodo: grave (se considera un error a cada omisión).

Fusionar en nivel medio baja de 933: moderado.

No reutilizar nodo 3 en el alta 400: grave.

Baja de 747 eliminar el nodo en vez de resolver el *underflow*: si fusionó en nivel medio en la baja de 933, no hay error, si no, inadmisible (no es algorítmico).

Escribir nodos que se liberan: moderado.

## Archivos Directos

0: 322, 504  
1: 162  
2: 359, 562  
3: 241, 598  
4: 480  
5: 281, -1  
6: 531, 650

DE=11/14

Las lecturas indicadas en rojo son para validar unicidad de claves.

+510: L6 L0 L1 E1

0: 322, 504  
1: 162, **510**  
2: 359, 562  
3: 241, 598  
4: 480  
5: 281, -1  
6: 531, 650

DE=12/14

-562: L2 E2

0: 322, 504  
1: 162, 510  
2: 359, **-1**  
3: 241, 598  
4: 480  
5: 281, -1  
6: 531, 650

DE=11/14

+301: L0 L2 **L4** E2

0: 322, 504  
1: 162, 510  
2: 359, **301**  
3: 241, 598  
4: 480  
5: 281, -1  
6: 531, 650

DE=12/14

+442: L1 L3 L5 **L0 L2 L4** E5

0: 322, 504  
1: 162, 510  
2: 359, 301  
3: 241, 598  
4: 480  
5: 281, **442**  
6: 531, 650

DE=13/14

## Errores posibles

Omitir la lectura o escritura de un bloque: grave se cuenta un error por cada omisión).

No verificar que haya claves duplicadas: grave.

Indicar una sola vez la densidad de empaquetamiento: leve.

Hacer mal una sola operación: grave.

No usar la segunda función en caso de colisión: inadmisibles.

Alta de 301, si inserta en bloque 4 en vez de bloque 2: grave.