IBBDD 2014 – Segunda Fecha de Primer Parcial – 8/7/2014

En la primer hoja del examen escribir claramente legajo, apellido y nombre, turno (mañana: 1 - tarde: 2), temas que rinde (por su número) y cantidad de hojas que entrega. Numerar cada hoja.

1. Archivos Secuenciales

El responsable de Sistemas de un negocio con varias sucursales obtiene mensualmente un archivo binario de cada una de ellas con el registro de facturas de ventas del mes pasado. Un programa que ya existe extrae de los registros de cada archivo de sucursal el **código de producto**, **cantidad vendida** y **precio de venta** de cada producto que aparezca en una factura, dejando esta información en otro archivo binario por cada sucursal, que luego ordena por código de producto.

Dadas las constantes y tipos de datos necesarios, codificar un procedimiento que reciba estos archivos de ventas mensuales ordenadas por producto (asignados y sin abrir) y, recorriéndolos una única vez, reporte en un archivo de texto (que también se recibe asignado y sin abrir) y **en una línea por cada producto**: el código de producto, la cantidad de unidades vendidas del mismo en cada sucursal (una columna por sucursal) y el total de pesos obtenido en el mes por ventas de ese producto (tener en cuenta que el precio de venta del mismo producto puede variar entre sucursales y entre registros de la misma sucursal). Se generaliza la solución para 5 sucursales, pero el procedimiento recibe el número real se sucursales con las que se trabaja. Codificar también la función para obtener el índice al mínimo código de producto.

```
Const CGSS = 5; {Cantidad General de Sucursales} MaxCod = 65535; {Máximo Código de Producto (inalcanzable)}

Type    tReg = Record codProd: Word; cant: Byte; pv: Real end;

        tArch = File of tReg; {ordenado por codProd y con repeticiones del mismo codProd}

        tCtrlArch = Record a: tArch; r: tReg; prodActual: Word end; {control de archivo de sucursal}

        tCtlMerge = Record
                crss: 1..CGSS; {cantidad real de sucursales}
                suc: Array[1..CGSS] of tCtlArch
                end;

Procedure Reporte( var ctl: tCtlMerge; var ventas: Text );
    Function min( var ctl: tCtlMerge ): Byte;
    var m, i: Byte;
    begin
        m:=1;
        for i:=2 to ctl.crss do if ctl.suc[i].r.codProd > ctl.suc[m].r.codProd then m:=i;
        min:=m
    end; {Min}


    Procedure leer( var a: tArch; var r: tReg );
    begin
        if eof(a) then r.codProd:=MaxCod else read(a, r)
    end;


var s, sm: Byte; {números de sucursal para recorrido y con codProd mínimo}
```

```
    codProdAct: Word;
    cant: Array[1..CGSS] of Word;
    totPesosPA: Real; {pesos producto actual}

begin {Reporte}
{Inicialización}
for s:=1 to ctl.crss do with ctl.suc[s] do begin
    reset(a); leer(a, r); prodActual:=r.codProd;
    end;
rewrite(ventas);
for s:=1 to ctl.crss do write(ventas, 'Suc.', s:2); {columnas de 6 espacios}
writeln(ventas, ' Total Pesos');
sm:=min(ctl); {índice de sucursal con codProd mínimo}

{Proceso}
while ctl.suc[sm].r.codProd<>maxCod do begin {ciclo de recorrido de productos}
    {inicialización producto actual}
    codProdAct:=ctl.suc[sm].r.codProd; totPesosPA:=0;
    for s:=1 to ctl.crss do with ctl.suc[s] do cant[s]:=0;
    while ctl.suc[sm].r.codProd=codProdAct do begin {ciclo de recorrido de sucursales con codProdAct}
            while ctl.suc[sm].r.codProd=ctl.suc[sm].prodActual do begin {ciclo de recorrido por sucursal
            para procesamiento del producto actual}
                cant[sm]:=cant[sm]+ctl.suc[sm].r.cant;
                totPesosPA:=totPesosPA+cant[sm]*ctl.suc[sm].r.pv;
                leer(ctl.suc[sm].a, ctl.suc[sm].r)
                end;
            {corte de control por sucursal}
            ctl.suc[sm].prodActual:=ctl.suc[sm].r.codProd;
            sm:=min(ctl);
            end;
    {corte de control por producto}
    for s:=1 to ctl.crss do write(ventas, cant[s]:6);
    writeln(ventas, totPesosPA:6:2)
    end;

{Finalización}
for s:=1 to ctl.crss do with ctl.suc[s] do close(a);
close(ventas)

end; {Reporte}
```

2. Árboles en Archivos

Dado el árbol B que se detalla más abajo, con capacidad máxima de 2 registros por nodo y mínima 1, muestre los estados sucesivos **completos** al realizar la siguiente secuencia de operaciones:

+500, -254, +800, -611.

Para cada operación indique la secuencia de nodos que debió leer o escribir. Considere que al partirse un nodo se agrega uno nuevo a la derecha, y si es hoja, el partido queda más cargado.

2: 0 (358) 3 (709) 1

0: (254) 3: (432)(611) 1: (724)(905)

+500:

6: 2 (500) 5			
2: 0 (358) 3		5: 4 (709) 1	
0: (254)	3: (432)	4: (611)	1: (724)(905)

L2L3E3E4E2E5E6

-254:

2: 0 (500) 4 (709) 1		
0: (358)(432)	4: (611)	1: (724)(905)

L6L2L0L3E0L5E2

Libres: 3, 5, 6

+800:

3: 2 (709) 5			
2: 0 (500) 4		5: 1 (800) 2	
0: (358)(432)	4: (611)	1: (724)	6: (905)

L2L1E1E6E2E5E3 (observar el orden en el que se reutilizan los nodos)

-611:

3: 2 (709) 5			
2: 0 (432) 4		5: 1 (800) 2	
0: (358)	4: (500)	1: (724)	6: (905)

L3L2L4L0E0E4E2

3. Archivos Directos

Se debe crear y cargar (sin comprobar unicidad de registros) un archivo directo de cubetas con capacidad para 2 registros con dispersión doble para organizar registros en saturación, con los 9 registros cuyas claves se listan a continuación y de manera que su densidad de empaquetamiento resulte del 75%: 347, 498, 729, 222, 113, 885, 431, 593, 709.

Usar como segunda función de dispersión el módulo 5 (resto de la división de la clave entre 5) más 1. Mostrar el estado del archivo luego de cada alta en la que el registro quede en saturación, y reportar para cada alta, las cubetas que se tuvieron que leer y escribir.

	Desborde	R1	R2
0	N	498	222
1	N	431	
2	N		
3	N	729	885
4	N		
5	S	347	113

347 mod 6 = 5: L5E5

498 mod 6 = 0: L0E0

729 mod 6 = 3: L3E3

222 mod 6 = 0: L0E0

113 mod 6 = 5: L5E5

885 mod 6 = 3: L3E3

431 mod 6 = 5; 431 mod 5 + 1 = 2, (5 + 2) mod 6 = 1: L5E5L1E1

	Desborde	R1	R2
0	N	498	222
1	N	431	593
2	N		
3	S	729	885
4	N	709	
5	S	347	113

593 mod 6 = 5; 593 mod 5 + 1 = 4, (5 + 4) mod 6 = 3; (3 + 4) mod 6 = 1: L5L3E3L1E1

	Desborde	R1	R2
0	S	498	222
1	S	431	593
2	N		
3	S	729	885
4	N	709	
5	S	347	113

709 mod 6 = 1; 709 mod 5 + 1 = 5, (1 + 5) mod 6 = 0; (0 + 5) mod 6 = 5; (5 + 5) mod 6 = 4: L1E1L0E0L5L4E4