

U.T.N. F.R.B.A. Algoritmos y Estructuras de Datos Prof.: Hugo A. Cuello
Guía de ejercicios Nro. 5
Estructuras datos dinámicas V, Recursividad, Templates, Árboles.

**Aplicar en los siguientes ejercicios, comprensión del problema,
diseño de la estrategia y desarrollar el algoritmo.**

1. Se tiene un archivo de Gastos producidos en un año calendario, contenido: Importe (real) y Mes (1..12), sin orden. Se requiere de un proceso que : Emite un listado de los gastos producidos en c/u. de los meses del año en orden inverso con respecto a como se conocieron c/u. de los datos. Además informar total por c/u. de los meses y el total anual.
2. Idem anterior, pero el listado deberá emitirse en el mismo orden con respecto a como se conocieron c/u. de los datos.
3. Idem al punto 1, pero el listado deberá emitirse ordenado por Importe en forma creciente.
4. Se tiene un archivo de Gastos producidos en un año calendario, contenido: Importe (real), Mes (1..12) y Día (1..31), sin orden, el cual podrá repetirse un gasto para un mismo mes de un mismo día. Se requiere de un proceso que : Emite un listado de los gastos producidos en c/u. de los meses del año y ordenado por Día (acumulado). Además informar total por c/u. de los meses y el total anual.
5. Crear un programa el cual solicite por medio de un menú apropiado, agregar, suprimir o listar los elementos de una lista.
6. Idem anterior, pero elementos de una pila.
7. Idem al punto 5., pero elementos de una cola.
8. Por medio de un menú apropiado que solicite Término del Polinomio 1, Término del Polinomio 2, Sumar Polinomios, Listar Polinomios; realizar las acciones correspondientes en un programa.
9. Dada una pila con elementos enteros generada en forma al azar, ordenarla en forma creciente y luego emitir los elementos. Usando otra pila
10. Dada una cola con elementos enteros generada en forma al azar, ordenarla en forma creciente y luego emitir los elementos. Usando otra cola.
11. Dada una expresión en notación infija, convertirla en notación prefija y luego emitirla.
12. Dada una expresión en notación infija, convertirla en notación postfija y luego emitirla.
13. Dado un archivo de datos desordenado, se solicita ordenarlo por el valor de un campo clave.
14. La D.G.I. requiere poder realizar el cobro sobre los Bienes Personales, para ello se cuenta con :
 - a) Un archivo de **Declaración Jurada** de las personas ordenado creciente por Apellido-Nombre, con el siguiente formato de registro: Apellido y Nombre (car.30), Dirección (car.30), Localidad (car.20), Código Postal (4 dígs.), Teléfono (car.15), Tipo y Nro. de documento (car.12), Importe Total de Bienes (real), originalmente en cero.
 - b) Un archivo de **Bienes Personales**, donde por cada persona habrá un registro de cabecera, seguido de uno o varios o ningún registro de bienes, cuyo formato de registro es : Tipo y Nro. de documento, Tipo de Bien (car.20), Importe (real),

Descripción (car.50). El archivo se encuentra ordenado creciente por Tipo y Nro. de documento y Tipo de Bien. En el registro cabecera el Tipo de Bien contiene un asterisco '*'. Cada registro del punto a) se corresponde con cada registro cabecera del punto b).

- c) Un archivo de Impuestos, no más de 10 registros, cuyo formato de registro contiene : Monto (real), Porcentaje (2 díg.), ordenado por Monto en forma creciente.

Se pide :

1. Un listado de c/u. de los bienes por cada persona, si el total de los valores es mayor o igual al monto mínimo, de acuerdo al siguiente diseño :

Listado de Bienes Personales que pagan Impuesto

Apellido, Nombres :	x-----x				
T-Nro.Documento :	x-----x				
T.Bien	Descripción	Valor	%	Impuesto	
x-----x	x-----x	99999,99			
		** Total :	99999,99	99	99999,99

2. Generar un archivo de Grandes Bienes de aquellas personas cuyo total de valores sea mayor al mayor monto establecido en Impuestos, con el siguiente diseño de registro : Apellido-Nombre, T-Nro.Doc., Importe Total.
 3. Actualizar el campo Importe Total de Bienes en el archivo del punto a).
15. Se requiere de un proceso que simule las siguientes acciones : Una ama de casa (doña Rosa), va al supermercado a realizar las distintas compras. Como suele olvidarse de las cosas que necesita, decide anotar cada requerimiento de productos en el orden en que surgen las necesidades. Al llegar al supermercado, comienza a realizar las compras, pasando por las góndolas, momento en que se detiene y observa si necesita algo de lo anotado; si es así, va tachando de su lista cuando se carga en el chango el producto encontrado. Esto continúa por cada góndola visitada. Para llevar a cabo estas acciones se cuenta con :
- a) un archivo de Artículos sin orden alguno cuyo formato de registro es : Código de Artículo (<=60000), Descripción (30 car.), Stock Actual (entero), Stock Mínimo (entero), Precio (real), Unidad de Medida (10 car.).
 - b) Una **lista de requerimientos de Compra** que deberá ser ingresada por el comprador antes de comenzar las compras, conteniendo como información : Descripción del Artículo (30 car), cantidad (entero).

Se pide :

1. Generar una lista de productos comprados (puede diferir de la lista de requerimientos), conteniendo como información lo siguiente : Cantidad (entero), dirección del Artículo en el archivo Artículos (entero largo).
2. Emitir el Ticket de la compra realizada, según el siguiente diseño :

Código de artículo,	
Descripción,	.
Cantidad,	Se repite por cada compra
Precio,	.
Total por artículo	

Cantidad de artículos : 999
 Total a Pagar : \$ 9.999,99

3. Emitir el Ranking de las 10 compras de artículos, de mayor importe total, ordenado decreciente por Importe, con el siguiente formato :

Ranking de los x artículos de mayor importe	
Código de Artículo	Importe
99999	99.999,99

4. Actualizar el stock del archivo artículos.
5. Generar un archivo de Reposiciónes, para aquellos artículos cuyo stock actual esté por debajo del stock mínimo, ordenado decreciente por la diferencia entre el stock mínimo y el stock actual, de acuerdo al siguiente formato de registro : Código de artículo, Stock Actual, Stock Mínimo.
16. Un Aeropuerto Internacional requiere las actualizaciones en el Tablero de Vuelos, para ello se cuenta con los siguientes archivos de datos :
- Empresas Aeronáuticas** : ordenado por Código de Empresa, máximo 20, conteniendo : Código de Empresa (5 car.), Nombre (30 car.), Domicilio (20 car.), Teléfono (15 car.), Otros datos (70 car.).
 - Vuelos** : (de un día, de 6:00 a 24:00 hs.), en donde el Nro. de vuelo se corresponde únicamente con la dirección del registro en el archivo, no más de 800 vuelos, conteniendo : Código de la Empresa, Código de Movimiento ('L' = Llega, 'S' = Sale), Lugar de Origen / Destino (30 car.), Hora (hhmm, de Llegada / Salida), Cantidad de Pasajeros (3 díg.), Nro. de Plazas (3 díg.), Escalas (lógico). El campo Cantidad de Pasajeros viene originalmente con valor cero, el proceso deberá actualizar la Cantidad de Pasajeros.
 - Pasajeros** : (Ordenado por Nro. de vuelo con repetición), conteniendo : Nro. de vuelo, Fecha (ddmmaa), Apellido y Nombre (20 car.), Edad (2 díg.), Nacionalidad (20 car.), Sexo ('F', 'M'), Tipo y Nro. de Documento (15 car.).

Se pide :

1. Visualizar el Tablero de Vuelos Programados para el día, de la siguiente forma:

Tablero de Vuelos del día : 99-99-99

Empresa	
Llegada Salida	

# Vuelo	# Vuelo
HH:MM	HH:MM
Origen	Destino

Empresa: x-----x	
Llegada	Salida

Cada vez que un vuelo llega o sale se deberá actualizar el tablero, el cuál puede contener un máximo de 10 vuelos a la vez.

- Mantener los vuelos (Llegadas, Salidas) en una estructura dinámica. (¿cuál será la más apropiada?).
- Emitir la Lista de Pasajeros del día de la fecha para todos los vuelos programados, según el siguiente diseño :

Lista de Pasajeros**# Vuelo :** 999**Empresa :** X(5)**¿Con escalas? :** XX**Llegada** HH : MM**Salida** : HH : MM**Apellido, Nombre****Nacionalidad****T.Nro.Doc.**

X(20)

X(20)

X(15)

Idem

Cantidad de Menores : 999**Cantidad de Mujeres :** 999**Cantidad Total :** 999**Nota :** HH:MM del nro.vuelo(1) < HH:MM del nro.vuelo(2) << HH:MM nro.vuelo(n).

El proceso controlará la hora del Sistema.

Si la edad es < 21 se considera menor de edad.

17. Un supermercado requiere un proceso que informe a diferentes consultas, contando para ello con :

1. Un archivo de **Artículos desordenado**, conteniendo: Código de Artículo (7 car.), Descripción del Artículo (30 car.), Stock Actual (entero), Stock Mínimo (entero), Código de Rubro (5 car.), Unidad de Medida (10 car.), Precio Unitario (real).
2. Un archivo de **Rubros, ordenado por Código de Rubro**, conteniendo : Código de Rubro, Descripción (30 car.).

Se pide :

1. Generar una “Lis de Rubros” ordenada, y por cada rubro generar “Listas de Artículos”, ordenada, que conforman dicho rubro, de acuerdo al siguiente diseño :
Por cada rubro : El código de rubro y dos punteros, uno al inicio a la lista de artículos y el otro al próximo rubro.
Por cada artículo : El código de artículo, la dirección del artículo y un puntero al próximo artículo.
2. Una vez hecho el punto 1. Informe :
 - 2.1. ¿Qué artículos pertenecen a cada rubro?, según el siguiente diseño:

Fecha : 99-99-99**Hoja :** 999**Listado por Rubros y Artículos****Cód.Rubro :** X(5) **Nombre :** X(30)**Cód.Art.** **Nombre****Precio**

X(7) X(30)

\$ 9.999,99

Cantidad de ítem : 9999

2.2. ¿Cuál es el Rubro que contiene la mayor cantidad de items?

2.3. Los valores porcentuales por Rubro :

Listado de valores porcentuales por Rubro**Cód.Rubro** **Valor %**

X(5) 999,99

3. Genere un archivo de índice Articulo.Idx, en el cual ordene lógicamente al archivo artículo, por Código de Rubro y dentro de éste por Código de Artículo, conteniendo : Clave y referencia en artículo.

18. Se requiere de un proceso que simule las siguientes acciones :

Se realiza una competencia automovilística por clasificación de distintas series. Por cada vuelta establecida deberá actualizarse las posiciones. Al término de la carrera, el ganador pasa a una “Lista de Clasificados”. Esto se repite por cada una de las series.

Al finalizar las series, se disputa la “Gran Final”. También en este caso por cada vuelta establecida debe actualizarse las posiciones. Al finalizar la competencia los tres primeros puestos suben al podio.

Para llevar a cabo estas acciones se cuenta con los siguientes datos :

- a) Un archivo de **Series**, con repetición ordenado creciente por Nro. de Serie, conteniendo : Nro. de Serie (entero), Nombre del Piloto (30 car.), Marca del auto (20 car.), Nro. del auto (byte).
- b) Un archivo de **Pilotos**, ordenado por Nombre del Piloto, conteniendo : Nombre del Piloto (30 car.), Lugar de Nacimiento (30 car.), Provincia (20 car.), Fecha de Nacimiento (ddmmaa), Puntaje en el Campeonato (entero), Cantidad de Carreras Ganadas en el Campeonato (entero).
- c) Se ingresa por teclado : Cantidad de vueltas en las “Series”, en la “Gran Final” y por cada vuelta y Piloto : Nombre del Piloto, Posición, Tiempo empleado, En el caso de producirse abandono, Nro. del auto.

Se pide :

1. Generar una “Lista de Carreras” -Series-, actualizando por cada vuelta las posiciones, en donde cada nodo debe contener como información : Nombre del Piloto, Nro. del auto, Tiempo de la vuelta más rápida. Indica la posición de largada el orden de aparición de los registros.
2. Generar una “Lista de Clasificados” -Gran Final-, de los ganadores de cada una de las series por orden de Tiempo de la vuelta más rápida.
3. Una vez finalizadas las series se disputa la Gran Final. Procesando la “Lista de Clasificados”, se debe actualizar por cada vuelta las posiciones, en donde cada nodo contiene la misma información que la “Lista de Carreras” del punto anterior.
4. Informar :
 - 4.1 Por cada serie : Nro. de Serie, Nombre del ganador, Nro. del auto, Mejor tiempo, la cantidad de participantes antes y después de la carrera.
 - 4.2 Los ganadores del punto 2. Para ocupar el podio (1er, 2do y 3er. puesto), con mensajes alcaratoiros y cantidad de abandonos.
 - 4.3 La lista de llegada del punto 2. :

Posición Nombre del Piloto Nro.auto Mejor tiempo

5. Actualizar el archivo Pilotos, para aquellos corredores que obtuvieron puntaje. El primer puesto también actualiza carreras ganadas.

Nota : Debe tomarse en cuenta los abandonos que se produzcan en cada competencia, eliminándolos de la lista.

El puntaje se establece de la siguiente manera : 1er. puesto: 40 puntos, 2do.puesto: 30 puntos, 3er.puesto: 15 puntos, 4to. puesto: 10 puntos y 5to.puesto: 5 puntos.

19. Dado un archivo de **Vacantes**, con la cantidad de vacantes de c/aula de la facultad, ordenado por Especialidad y División, donde c/registro contiene :

Nro. de Especialidad (1 a 9), Nro. de División (1 <= 40, 41 <= 60, 61 <= 99), Cantidad de Vacantes (<=99), Cantidad de Inscriptos (<= 99 originalmente en cero).

Existe al menos una división por cada turno, en todas las especialidades.

Se dispone además de un archivo de **Solicitudes**, en orden cronológico de los alumnos de primer año que solicitan inscribirse, donde c/registro contiene : Nro. de Legajo (6

díg.), Apellido y Nombres (30 car.), Nro. de Especialidad, Código de Turno ('M', 'T', 'N').

Se pide :

1. Asigne división a cada solicitante.
2. Grabe un registro, con el formato del 2do. Archivo más un campo de Nro. de División, por c/u. de los registros leídos del mismo archivo, para aquellos solicitantes que no encontraron vacante, este campo debe contener cero.
3. Haga un listado de los alumnos solicitantes en c/división de c/especialidad, ordenado por Especialidad, Turno y Cronológico, según se indica :

Especialidad : 9	Turno : x---x
Orden Original	Legajo Apellido, Nombre
99	999999 x-----x

Nota : Se dispone de 216 bytes en memoria estática. Tamaño de nodos 8 bytes, con un nodo por cada solicitante.

20. Una municipalidad requiere un programa que permita realizar la concelación de deuda de sus contribuyentes. Para ello se dispone de los siguientes archivos :
 - a) **Deudores sin ningún orden**, donde c/registro contiene : Código de Contribuyente (5 car.), Año de la cuota (88 a 97), Mes de la Cuota (1 a 12), Importe de la Cuota, Estado ('A' = Adeuda, 'C' = Cancelada), Fecha de Cobro (aammdd).
 - b) **Datos Personales** (<= 8000 registros ordenados por Contribuyente), c/registro con : Código de Contribuyente, Apellido y Nombre (45 car.), Domicilio (45 car.), Código de Zona Municipal (4 car.).
 - c) **Cobro del Mes ordenado por Fecha de Cobro y Código de Contribuyente**, donde c/registro contiene : Código de Contribuyente, Año de la Cuota, Mes de la Cuota, Fecha de Cobro (aammdd).

Se pide :

1. Actualice en el archivo Deudores las cuotas pagadas por cada contribuyente.
2. Imprima un Listado para control de cobranza como se indica :

Listado de Control de Cobranza

Contrib.	Apellido, Nombre	Año	Mes	Importe	Situac.
X(5)	X(45)	99	99	99999,99	P
		99	99	99999,99	E

En Situación se lo indica con **P** = **Pago** aceptado o **E** = **error** por cuota pagada anteriormente.

3. Imprima el total de lo recaudado por c/zona municipal ordenado por código de zona:

Código de Zona	Importe Recaudado
xxxx	999999,99

Nota : Se dispone de 4 Mbytes de memoria.

21. La Capital Federal requiere un estudio estadístico de infracciones realizadas por todo tipo de vehículo en sus calles. Para ello cuenta con 2 archivos c/u de ellos con infracciones de los 12 meses del año 1997, correspondientes a las 2 zonas en que se encuentra dividida la capital, para los que c/registro contiene: Cód. de patente (6 car), Nro. registro Conductor (8 díg.), Fecha de la infracción (aammdd), Zona (6 car.), Cód. de la infracción (5 dig), Nro. placa policial (6 car.). Ambos ordenados por fecha creciente.

Se pide :

1. Genere un único archivo: **CAPITOT.DAT** a partir de los 2 mencionados, ordenado por código de patente (creciente) y fecha (decreciente).
2. Para los vehículos con infracciones en todos los meses del año, imprima el siguiente listado :

Total de Infracciones por mes

COD/PATENTE	Mes 1	Mes 2	Mes 3	Mes 11	Mes 12	Total
XXXXXX	9999	9999	9999		9999	9999	99999

NOTA: Tamaño máximo de nodo 15 bytes. Memoria estática 0,1 Kb.

22. En una Universidad se requiere consolidar la información de aprobación de trabajos prácticos de una de sus especialidades. Para ello se cuenta con los siguientes archivos :
 - a) Un archivo de **Asistencias regulares** generado por bedelía, ordenado por Materia, donde cada registro contiene : Nro. de Legajo (6 díg.), Código de Materia (3 díg.). Solo están los alumnos que terminaron de cursar c/regulares.
 - b) Un archivo de **Notas de parciales** generado por los docentes, ordenado por número de curso, donde c/registro contiene : Nro. de Curso (2 díg.), Código de Materia, Nro. de Legajo, Nota primer parcial (0 a 10), nota segundo parcial (0 a 10), Fecha de firma T.P. (ddmmaa). Nota : Están todos los alumnos inscriptos originalmente, una fecha = 0 indica que no firmó.
 - c) Un archivo de **Trabajos Prácticos**, previamente aprobados, ordenado por fecha de firma, donde cada registro contiene : Nro. de Legajo, Código de Materia, Fecha de firma, Nro. de Curso.

Se pide :

1. Detecte e imprima los 3 posibles errores del archivo de Notas (algún parcial desaprobado, o sin fecha de firma o sin regularidad).
2. Genere un archivo de Trabajos Prácticos aprobados, ordenado por legajo y código de materia, a partir del archivo del punto c) y de los registros correctos del archivo del punto b).

Nota : Tamaño máximo del nodo 13 bytes; máximo espacio utilizable en disco = filesize(Notas) + filesize (TrabPrac).

23. La empresa MoviTel necesita de un proceso para facturar a sus clientes las comunicaciones realizadas en un mes. Para ello cuenta con los siguientes archivos de datos:
 - a) **COMUNICACIONES** (ordenado por Cliente), conteniendo el registro: Número de Cliente (9 díg.); Cantidad de pulsos (3 díg.); Día del mes (1 a 31).
 - b) **CLIENTES** (ordenado por Cliente), conteniendo: Número de Cliente; Razón Social (35 car.); Domicilio-Localidad (50 car.); Categoría (2 díg.).
 - c) **CATEGORIAS** (sin orden), conteniendo: Código de Categoría; Importe de abono; Importe por pulso adicional; Pulsos sin cargo (3 díg.).

Se pide:

1. Imprima la factura de cada cliente (ordenado por Cliente), según el siguiente formato:

Nro.Cliente: 9(9)	Razón Social: X(35)	
Domicilio: X(50)		
ABONO	(9999 pulsos sin cargo)	999.99
ADICIONAL	(9999 x imp.pulso-adic.)	+ 999.99
T O T A L		9999.99

2. Imprima al final una lista (ordenada por **categoría y cliente**) de los clientes que tuvieron menos de 50 pulsos en alguno de los días, con el siguiente formato:

LISTADO DE COMUNICACIONES DIARIAS

Categoría: 99

Nro.Cliente	1	2	3	<<	>	29	30	31	Cant.Dias Pul.< 50
999999999	999	999	999			999	999	999	99

NOTA: La memoria disponible es suficiente para estructuras dinámicas si se utiliza un nodo por cada usuario que cumpla la condición del punto 2. No olvidar de liberar la memoria cuando ya no se necesite un nodo.

Realizar gráfico de estructuras aclarar que tipo de estructura lineal se emplea, estructura del nodo, cantidad de memoria a emplear, definir y declarar la sección type y var, desarrollar estrategia, algoritmo y codificación en Pascal. Minimizar el tamaño de los nodos.

24. “**AEROPUERTOS XXI**” requiere de un proceso de los vuelos de un día desde y hacia diferentes destinos, contando con los siguientes archivos :

- a) **PASAJEROS**, ord. x Nro.Vuelo del día., conteniendo : Nro.Vuelo (5 díg.), Apel.Nom.(20 car.), Nro.DNI (8 díg.), Nacional.(20 car.), Fec.Nac.(8 díg. aaaammdd), Clase (0='Especial',1='Primera',2='Común'), Sexo ('F', 'M').
- b) **VUELOS**, con 100.000 registros, en donde: Nro.de Vuelo = dir.en el archivo, conteniendo : Nro.de Vuelo, Cód.Aerop.Origen, Cód.Aerop.Destino, Hora Salida (4 díg. = hhmm), Tiempo (4 díg. = hhmm).

Nota: Nro.Vuelo = 0, significa que no es un vuelo válido para el día. El intervalo de tiempo entre un vuelo y otro no es menor a 60 segundos y no existe posibilidad de solaparse. El primer vuelo llega/sale a las 00:00 y el último vuelo llega/sale a las 23:59 de existir un vuelo en esos horarios por cada aeropuerto.

- c) **AEROPUERTOS**, con 100 regs. Relac. 1:1 Cód.Aerop. = dir.arch., conteniendo: Nom.Aerop. (20 car.).
- d) **TARIFAS**, de cada clase y destinos (real x 3), sin repetir el mismo par de aeropuertos, ordenado de 0-1 a 0-99, 1-2 a 1-99, ... , 98-99.

Se pide :

1. Emitir un listado de las **Salidas** del día, de todos los vuelos de “**AEROPUERTOS XXI**”, ord. x Cód.Aerop. creciente y Nro.Vuelo decreciente de las Salidas de todos los aeropuertos:

SALIDAS DE VUELOS

CÓD.AEROP.: 99 AEROP. : X(20) (Origen)

**** CÓD.AEROP.: 99 AEROP. :** X(20) (Destino)

**** NRO.DE VUELO:** 99999 **HORA:** hh:mm (de salida)

ORDEN	APELLIDO, NOMBRE	Nro. Doc.	Fec. Nac.
999	XXXXXXXXXXXXXX XX	99.999.999	99/99/9999

2. Grabar en un archivo de **Llegadas** del día, de todos los vuelos de cada aeropuerto con igual orden que el punto 1., con el siguiente formato de registro: Cód.Aero.Orig, CódAero.Dest., Nro.Vuelo, Hora de llegada (hhmm), Cant.Hombres, Cant.Mujeres, Importe de cada clase.

RESTRICCIONES:

Memoria estática: 40.000 bytes. **Memoria en disco:** Solo para el punto 2. **Memoria dinámica:** 1 nodo por cada Nro.Vuelo sin repetir. **Tamaño del nodo:** 8 bytes.

25. Una empresa de turismo representante de una compañía aeronáutica, para gestionar sus viajes en el mes de enero, reserva con anticipación una cierta cantidad de pasajes aéreos a diferentes ciudades del mundo (máximo un vuelo por ciudad/día). Se cuenta con los siguientes archivos :

- RESERVAS** (sin orden), donde cada registro contiene: Código de Ciudad (3 díg.), Día de Salida (2 díg.), Hora de Salida (6 díg.), Total de Pasajes Reservados (3 díg.), Nro. de Vuelo (4 díg.), Precio de cada Pasaje (real).
- PEDIDOS** (sin orden), con los pasajes solicitados por sus clientes, conteniendo : Código de Ciudad, Cantidad Solicitada (3 díg.), Día de Salida, Nro. de Documento (8 díg.).

Desarrollar la estrategia, algoritmo y codificación del programa que :

1. Imprima para el día en que se rechazaron más pedidos, el siguiente listado de rechazos (ordenado por Código de Ciudad) :

DIA DE MAXIMO RECHAZO 99		
NRO. DOCUMENTO	COD. CIUDAD	CANT. SOLICITADA
99999999	9999	999

2. Grabe un archivo **VENTAS.DAT** con los pedidos aceptados (registro de igual tipo que el de PEDIDOS más un campo Importe -resultado de Cant. Solicitada * Precio-, ordenado por día y Cód. de Ciudad).

NOTA :

Memoria estática: disponible 309 Kb.

Memoria dinámica: fileSize(Pedidos) * 8. Tamaño del nodo 8 bytes.

Espacio en disco : utilizar menos que el utilizado por PEDIDOS.DAT * 1,6.

26. Una empresa de ventas domiciliaria de electrodomésticos que requiere conocer el desempeño de sus vendedores y clientes, cuenta con :

- a) **VENTAS** (ordenado por mes/día de venta), con las ventas del último año donde c/registro contiene:

Nro.de Cliente (7 díg.), Mes/día de venta (4 díg.), Código de Vendedor (6 car.), Importe de la Venta.

- b) **CLIENTES** (ordenado por Nro. de Cliente), donde c/registro contiene :

Nro. de Cliente, Domicilio (40 car.), Apellido y Nombre (30 car.), Nro. de CUIT (11 díg.).

Desarrollar la estrategia, algoritmo y codificación del programa que :

1. Imprima un listado de ventas mensuales de cada vendedor ordenado por código de vendedor, mes y día (pero sólo para los meses en los cuales el importe total del mes supera al promedio general del mes).

VENDEDOR : X(6)	MES : X(10)
DIA	APELLIDO, NOMBRE -del cliente- IMPORTE
99 X(30)	999999.99
	IMPORTE TOTAL : 9999999.99

2. Un listado de los clientes que le compraron a dos o más vendedores según se indica :

APELLIDO, NOMBRE -del cliente-	IMPORTE TOTAL
X(30)	9999999.99

NOTA : Los vendedores son 200 como máximo.

Se dispone de 98 kb. De memoria. Tamaño máximo de nodo = 8 bytes.

Máximo espacio disponible en disco filesize(Clientes) * 14 bytes.

27. Una empresa prestadora de servicios de internet, requiere facturar a sus usuarios y para ello dispone de los siguientes archivos :

- CONEXIONES** (orden cronológico), en el que cada registro contiene : Día del mes (1 a 31), Número de usuario (8 dígs.), Código de destino (16 car.), Cantidad de minutos (3 dígs.).
- CATEGORIA** (sin orden), en el que cada registro contiene : Categoría, Minutos sin cargo (4 dígs.), Importe de abono, Importe por minuto adicional.
- USUARIOS** (ordenado por usuario), en el que cada registro contiene : Número de usuario, Apellido y Nombre (34 car.), Domicilio (49 car.), Categoría (2 dígs.).

Desarrollar la estrategia, algoritmo y codificación del programa que :

1. Imprima la factura de cada usuario (ordenado por su número), según se indica:

USUARIO NRO.: 99999999	APELLIDO Y NOMBRE : X(34)
-------------------------------	----------------------------------

DOMICILIO : X(49)	
	ABONO (999 minutos sin cargo) 999.99
	ADICIONAL 999 x imp.adic. + 999.99
	TOTAL 9999.99

2. Imprima ordenado por código de destino y día todas las conexiones individuales de usuarios que duraron más de 360 minutos, según se indica:

CODIGO DE DESTINO: X(16)	
DIA	APELLIDO Y NOMBRE USUARIO
99	MINUTOS X(34) 9999

NOTA : Hay 130 Kb. de memoria estática. Nodo máximo = 8 bytes. No hay espacio en disco. Cantidad máxima de destinos 5000.

Recursividad

En todos los casos se solicita crear una función para:

1. Hallar el factorial de un número n.
2. Obtener el M.C.D.
3. Encontrar el valor del término n de la sucesión de Fibonacci.
4. Emitir los elementos de una lista.
5. Determinar si una cadena dada es o no un palíndromo.
6. Comprobar si un valor clave se encuentra o no en un array ordenado por el valor de esa clave, retornando la posición en donde se encontró, caso contrario, retornar cero.
7. Calcular la potencia de a elevado a la n.