EJERCICIOS MÓDULO III

La solución de los ejercicios consiste en el desarrollo de los pasos que mencionamos a continuación:

1. Comprensión del enunciado
2. Especificación de los datos de entrada y de salida
3. Confección de la estrategia de solución (top-down)
4. Confección del algoritmo de resolución (refinamientos sucesivos)
5. Seguimiento del algoritmo
6. Codificación
7. Compilación
8. Preparación del lote de prueba
9. Ejecución y depuración del programa

PUNTEROS

1. Declare una variable puntero a un entero llamada **NUMERO**, pida memoria para ella, asigne el valor 10 en la posición de memoria que contiene esta variable puntero. Luego muestre el contenido de la porción de memoria a la cual apunta **NUMERO** por pantalla. Finalmente libere la porción memoria a la que apunta **NUMERO**.

2. Sume a la posición de memoria a la cual apunta la variable del ejercicio anterior el valor 21 y luego muestre el resultado por pantalla.

3. Asigne memoria para tres variables reales, pida al usuario que ingrese valores para dos de esas tres variables, y luego muestre por pantalla la suma, diferencia, producto y cociente entre ellas, utilizando como variable auxiliar la tercer porción de memoria que reservó.

4. Analice el siguiente fragmento de código, e indique si las operaciones realizadas son correctas o no y por qué:

```
VAR
  Ptr1,Ptr2: ^byte;
BEGIN
   New(Ptr1);
   New(Ptr2);
a) Ptr1 := 2;
b) Ptr^ := 2;
c) Ptr1 := 2;
d) Ptr2 := Ptr1;
e) Ptr2 : = Ptr1^;
f) Ptr2^ := Ptr1^;
g) Ptr2^ := Ptr1;
```

5. Encuentre los errores en el siguiente código, corríjalos y diagrame el algoritmo correspondiente a su corrección

```
VAR
  Ptr1,Ptr2:byte;
BEGIN
  New(Ptr1);
  New(Ptr2);
  Ptr1 := 10;
  Ptr2^ := 5;
  Writeln('Los valores almacenados en memoria son ',Ptr1,' y ',Ptr2);
  Ptr1^ := Ptr2^ + 3;
  Writeln('Ahora los valores son ',Ptr1^,' y ', Ptr2^);
  Ptr2^ := Ptr1;
  Writeln('En este momento  ambos punteros apuntan a la misma dirección de memoria');
  Dispose(Ptr1);
  Dispose(Ptr2);
  Writeln('Ya liberé la memoria a la que apuntaban los punteros');
  Ptr1^ := Nil;    Ptr2^ := Nil;
```

Writeln('Es una buena práctica dejar un puntero apuntando a nada luego de haber liberado la memoria') **END.**

6. Declare una variable puntero a un string de 30, y a continuación cargue en la región de memoria a la cual apunta la cadena ´Es una cadena de caracteres´. Muestre el contenido de esta variable en pantalla.
7. ¿Es posible tener una variable que apunte a una variable puntero? Justifique.
8. Declare dos variables puntero que apunten a valores de tipo integer. En la primera región de memoria guardará la característica de un número telefónico, y en la segunda, el número de usuario, ambos valores ingresados por teclado. A continuación muestre el número completo en formato 4999-9999.
9. Declare una variable de tipo puntero al siguiente registro:

NUMERO	LETRAS
1 dígito	6 caracteres

   Luego pida memoria, asígnele un valor a cada campo del registro, y muéstrelo por pantalla.
10. Declare un puntero a una matriz de bytes de 16 filas por 16 columnas, y a continuación rellénela con los valores posibles para esta variable. Muestre por pantalla su contenido. Devuelva la memoria utilizada por la matriz.
11. Cree una estructura que le permita almacenar hasta 100 cadenas de caracteres de longitud máxima (255 caracteres), teniendo como restricción sólo *400 bytes de memoria estática*.
12. Modifique los algoritmos de Búsqueda Binaria en un vector, y de Ordenamiento de vectores, para que reciban como parámetro un puntero a un vector en lugar del vector mismo.
13. Diseñe un algoritmo que lea desde un archivo **VALORES.DAT** 100 valores reales y los muestre en pantalla ordenados de mayor a menor o a la inversa, según desee el usuario. Utilice un vector en memoria dinámica para resolverlo.

LISTAS ENLAZADAS

14. Resuelva el ejercicio 13 utilizando listas enlazadas. Analice las ventajas y desventajas entre ambas soluciones.
15. Cree un algoritmo que le permita obtener un archivo ordenado por **NOMBRE_DE_USUARIO** con el siguiente formato:

NOMBRE_DE_USUARIO	CONTRASENIA
10 caracteres	8 caracteres

    Los datos se cargan por teclado sin orden preestablecido.

    Nota: No hay memoria para almacenar estructuras en memoria estática y la memoria dinámica es suficiente.

16. A partir del siguiente archivo:

    TIEMPO.DAT, donde cada registro contiene:

CORREDOR	TRAYECTO	TIEMPO
3 dígitos	2 dígitos	4 dígitos

    Se pide que imprima un listado ordenado por Corredor, indicando su trayecto y tiempo.

    Nota: No hay memoria estática disponible, la memoria dinámica es suficiente.

17. Dado un archivo **MATERIAS.DAT** (sin orden) que registra la cantidad de inscriptos a un final:

DIVISION	ESPECIALIDAD	TURNO	CANT. INSCRIPTOS
3 dígitos	2 dígitos	1 carácter	3 dígitos

    Se pide un listado por **ESPECIALIDAD**, indicando la cantidad de divisiones e inscriptos.

    Nota: No se cuenta con memoria estática, la memoria dinámica es suficiente.

18. A partir del siguiente archivo:

VENTAS.DAT (sin orden)

FACTURA	COD_VENDEDOR	COD_CLIENTE	MONTO
4 dígitos	3 dígitos	10 caracteres	5 dígitos y 2 decimales

Se pide imprimir un listado ordenado por **FACTURA**, indicando cada uno de los distintos ítems de la compra. Realizar lo pedido considerando las siguientes restricciones:

a) Memoria suficiente para un nodo máximo de 25 bytes
b) *Memoria suficiente para un nodo por factura, nodo máximo de 8 bytes*.

Nota: *No hay memoria estática.*

19. Se provee el siguiente archivo:

EXPEDIE.DAT (ordenado por **DNI**)

DNI	NOMBRE	FECHAINIC	JUZGADO	FECHAREV	EXPEDIENTE	JUEZ	SENTENCIA
8 dígitos	30 caracteres	AAAAMMDD	4 dígitos	AAAAMMDD	5 dígitos	4 dígitos	30 caracteres

Se pide un listado de aquellos expedientes que aún no tengan sentencia, ordenado por **FECHAINIC**.

Nota: El campo **SENTENCIA** en blanco indica que aún no tiene sentencia.
Sólo se dispone de memoria dinámica, la cual alcanzará para resolver la tarea sólo si los nodos no superan 12 bytes de tamaño.

20. Se cuenta con el archivo **EMPRESAS.DAT**, sin orden, y con el siguiente formato de registros:

DENOMINACION	ZONA	DIRECCION	TELEFONO
30 caracteres	1 dígito	30 caracteres	12 caracteres

Se pide que genere un nuevo archivo llamado **ORDENADO.DAT**, con el mismo formato de registro y ordenado en forma creciente por el campo **DENOMINACION**, según las siguientes pautas:

a) Sin memoria para estructuras estáticas, 416.000 bytes de memoria dinámica.
b) Sin memoria para estructuras estáticas, nodo de 37 bytes como máximo.

Nota: *El archivo **EMPRESAS.DAT** tiene como máximo 5200 registros.*

21. Se cuenta con el archivo **OPERA.DAT**, sin orden, con la siguiente estructura:

CODIGO	MONTO
5 caracteres	4 dígitos y 2 decimales

Se pide que imprima un listado ordenado por **CODIGO** creciente y otro por **MONTO** decreciente.

22. Se cuenta con el archivo **SEGUROS.DAT**, desordenado, cuyo formato de registro es el siguiente:

POLIZA	FECHA	DURACION	ASEGURADO	PRODUCTOR	COBERTURA	PATENTE	MONTO
5 dígitos	AAAAMMDD	3 dígitos	30 caracteres	6 dígitos	3 dígitos	8 caracteres	3 dígitos y 2 caracteres

Se pide que verifique cuantas pólizas posee cada asegurado y lo informe en la pantalla, teniendo en cuenta que no hay más de 5500 pólizas, pero que un asegurado puede tener más de una.

También debe imprimir un ranking con los 5 mejores productores, sean aquellos con mayor cantidad de pólizas vendidas, en orden decreciente.

Nota*: No hay memoria estática, la memoria dinámica es suficiente.*

23. A partir del siguiente archivo:

VUELOS.DAT (desordenado)

NVUELO	EMPRESA	ORIGEN	DESTINO	HORASALIDA	HORALLEGADA	PASAJEROS
4 dígitos	10 caracteres	20 caracteres	20 caracteres	HHMM	HHMM	3 dígitos

Se pide un listado de vuelos, con todos los campos, ordenado por NVUELO y un listado de empresas, ordenado por cantidad de vuelos decreciente.

Nota: La cantidad máxima de empresas es de 50 y de vuelos por empresa es de 20.
No se dispone de memoria estática, La memoria dinámica disponible es de 15 Kbytes.

24. Dado el archivo **CURSOALU.DAT**, ordenado por **LEGAJO** y donde cada registro tiene el siguiente formato:

APELLIDO_Y_NOMBRE	LEGAJO	DIVISION
35 caracteres	7 dígitos	4 caracteres

Desarrollar el algoritmo y codificación del programa que imprima:

a) Un listado de alumnos por división, ordenado por **DIVISIÓN** y **LEGAJO** crecientes.
b) Un listado de alumnos por división, ordenado en forma creciente por **DIVISIÓN** y decreciente por **LEGAJO**.

Nota: Las divisiones asignadas no son más de 100.
La memoria estática disponible es 950 bytes y la memoria dinámica es suficiente.

25. Se dispone de los siguientes archivos:

USUARIOS.DAT (sin orden)

USUARIO	NOMBRE	NICKNAME	PASSWORD	CLASE
4 dígitos	30 caracteres	10 caracteres	10 caracteres	3 dígitos

ACCESO.DAT (sin orden)

FECHA	HORA	DURACION	USUARIO	SERVER	IPASIGNADO
AAAAMMDD	HHMM	HHMM	4 dígitos	3 dígitos	7 caracteres

Se pide imprimir un listado ordenado por **USUARIO**, donde se indicará: el **NOMBRE** y **NICKNAME** del **USUARIO**, los datos pertenecientes a cada uno de los distintos accesos efectuados y finalmente el tiempo total utilizado.
Al terminar el proceso indicar toda la información sobre el usuario o usuarios con mayor consumo.

Nota: *No hay memoria estática.*
No hay más de 60000 accesos, ni más de 1000 usuarios.
La memoria dinámica alcanza sólo si el nodo no tiene más de 12 bytes.

26. A partir del siguiente archivo:

COMPACTS.DAT (sin orden)

CODIGOCD	CANCION	INTERPRETE	COMPOSITOR	MINUTOSCANCION
9 dígitos	2 dígitos	7 dígitos	6 dígitos	2 dígitos

Se pide un listado ordenado por **CODIGOCD**, indicando la cantidad de canciones, la cantidad de intérpretes y la duración del CD .

Nota: *Existe un máximo de 1000 CDs.*
a) Se cuenta con 0 bytes de memoria estática y memoria dinámica suficiente
b) Se cuenta con 11 Kb. de memoria estática y memoria dinámica suficiente.

27. Se cuenta con los siguientes archivos:

CLIENTES.DAT (sin orden)

NOMBRE_Y_APELLIDO	EDAD	FECHA_NAC	DNI	DEUDA
30 caracteres	2 dígitos	8 dígitos	8 dígitos	4 dígitos y 2 decimales

PAGOS.DAT (ordenado por **DNI**)

DNI	FECHA	MONTO
8 dígitos	AAAAMMDD	4 dígitos y dos decimales

Se pide que imprima un listado ordenado por **DNI** que muestre su deuda actualizada.

Nota: *Se sabe que no hay más de 12800 clientes, y que un cliente puede haber efectuado más de un pago.*
Se dispone de 180 Kb de memoria dinámica y 0 bytes de memoria estática.

28. Realice un algoritmo y su codificación que le permita obtener un archivo con el siguiente formato:

    ALUMNOS.DAT (ordenado por **NUMLEG** creciente y **CODMAT** decreciente)

NUMLEG	CODMAT	NOTACURSADA	NOTAFINAL
7 dígitos	6 dígitos	2 dígitos	2 dígitos

    Los datos se cargarán por teclado, sin ningún orden preestablecido y finalizan con un **NUMLEG** igual a **0**.

    Nota: *No hay memoria estática disponible, y se dispone de memoria suficiente sólo si el tamaño del nodo no supera los 12 bytes.*

29. Dado un archivo de registros de alumnos **ARCHA** sin ningún orden donde cada registro contiene:

APELLIDOY NOMBRE	NUMERODELEGAJO	DIVISIONASIGNADA
34 caracteres	6 dígitos	3 dígitos

    Se debe desarrollar el algoritmo y codificación del programa que imprima el listado de alumnos por división, ordenado por **DIVISIONASIGNADA** y **NUMERODELLEGAJO** crecientes

    Considerar las siguientes alternativas:

    a) Memoria estática 3 Kb; dinámica suficiente si ningún nodo ocupa más de 14 bytes.
    b) Memoria estática 4 Kb; dinámica suficiente si ningún nodo ocupa más de 12 bytes.
    c) Memoria estática 3 Kb; dinámica suficiente si ningún nodo ocupa más de 12 bytes

30. Igual planteo que el Ejercicio anterior, pero con dos archivos de alumnos sin orden **ARCHA** y **ARCHB**.

    Considerar las siguientes alternativas:

    a) Memoria estática 1 Kb; dinámica suficiente si ningún nodo ocupa más de 15 bytes
    b) Memoria estática 8 Kb; dinámica suficiente si ningún nodo ocupa más de 12 bytes
    c) Memoria estática 2 Kb; dinámica suficiente si ningún nodo ocupa más de 13 bytes

31. Dada una lista (nodo = registro + puntero), desarrollar y codificar una función que devuelva la cantidad de nodos que tiene.

32. Dada una lista **LISTA** (estructura dinámica), imprímala en el orden en que se encuentra si contiene más de 100 elementos, y en caso contrario, en orden invertido.

33. Dadas dos listas **LISTA** y **LISTB** (estructura dinámica), ambas del mismo tipo, y ordenadas en forma creciente por el campo **VALOR**, desarrollar y codificar un procedimiento que genere una única lista **LISTC** a partir de ellas, también ordenada.

PILAS Y COLAS

34. Desarrollar un procedimiento que ingrese por teclado un conjunto de Apellidos y Nombres de alumnos, finalizado con un Apellido vacío, y que los imprima en el orden inverso al ingresado.

35. Definir una función **INVERSA** que evalúe dos conjuntos de caracteres separados por un punto (*por ejemplo AzByCx.xCyBzA*), que devuelva TRUE si uno de los conjuntos es la reversa del otro o False si no lo son. Los datos deberán ingresarse por teclado, y *no puede utilizarse memoria estática*

36. Se cuenta con el siguiente archivo:

    LEGAJOS.DAT (ordenado por **NUMERO_LEGAJO** decreciente)

NUMERO_LEGAJO	CANTIDAD_MATERIAS
9 dígitos	2 dígitos

    Se pide obtener un listado ordenado por **NUMERO_LEGAJO** creciente.

37. A partir del siguiente archivo:

CUOTAS.DAT (ordenado por **CLIENTE**)

NUMCLI	NUMCUOTA	FECHAPAGO	FECHAVENC	MONTOABONADO
3 dígitos	2 dígitos	AAAAMMDD	AAAAMMDD	4 dígitos y 2 decimales

Se pide que cree un nuevo archivo **DUPLICAD.DAT**, con el orden inverso al que tiene **CUOTAS.DAT**, recorriendo este último sólo en forma secuencial de principio a fin, y sin utilizar memoria estática.

38. Se cuenta con el siguiente archivo:

CLIENTES.DAT (ordenado por **CODCLIENTE**)

CODCLIENTE	NOMBRE	EDAD	FECHANAC	DNI	TELEFONO
8 dígitos	30 caracteres	2 dígitos	AAAAMMDD	7 dígitos	13 caracteres

Se pide que imprima un listado de clientes con todos sus datos en orden inverso al del archivo, teniendo en cuenta las siguientes restricciones:

a) No hay memoria estática, la memoria dinámica alcanza para cargar todos los datos del cliente en cada nodo.
b) No hay memoria estática y sólo se dispone de memoria dinámica suficiente si el nodo no tiene más de 8 bytes.

39. A partir del archivo **DEUDORES.DAT**, ordenado en forma decreciente por los campos **APELLIDO** y **NOMBRE**, cuyo formato de registros es el siguiente:

APELLIDO	NOMBRE	NCREDITO	ULTIMOPAGO	DEUDA
15 caracteres	15 caracteres	4 dígitos	AAAAMMDD	4 dígitos y 2 decimales

Deberá imprimirse un listado, ordenado alfabéticamente por **APELLIDO** y **NOMBRE**, conteniendo todos los datos de aquellas personas que no hayan pagado desde una fecha que el usuario ingresará por teclado.

Nota: *Sólo se dispone de memoria dinámica, la cual alcanzará para la tarea si cada nodo no supera los 6 bytes. El archivo no tendrá más de 65000 registros.*

40. Dadas dos pilas **PILA** y **PILB** (estructura dinámica), ambas del mismo tipo, y ordenadas en forma creciente por el campo **VALOR**, desarrollar y codificar un procedimiento que genere una única pila **PILC** a partir de ellas, también ordenada por el mismo campo.

41. Se provee del siguiente archivo, perteneciente a los registros de alumnos de una escuela polimodal privada:

ALUMNOS.DAT (ordenado por **DNI** en forma descendente)

NINSCRIPC	NOMBREYAPELLIDO	DNI	GRADO	DIVISION	MONTOCUOTA	PORCENTAJEBECA
4 dígitos	30 caracteres	8 dígitos	1 dígito	1 carácter	3 dígitos y 2 decimales	3 dígitos

Se solicita un programa que sea capaz de imprimir un listado de los alumnos que reciban beca (**PORCENTAJE** > 0), discriminado por **GRADO** y ordenado por **DNI** creciente, según el siguiente modelo:

GRADO: 9

DNI	División	Cuota	Beca	Total a abonar
99999999	A	$ 999.99	100 %	$ 999.99

Nota: *Sólo hay 37 bytes de memoria estática, la memoria dinámica es suficiente si no se usan más de 16 bytes por nodo.*

42. Se cuenta con los siguientes archivos:

REPOSITOR.DAT (desordenado)

CODREPOS	DNI	NOMBREYAPELLIDO	PREMIOS	EXTERNO
4 dígitos	8 dígitos	25 caracteres	2 digitos y 2 decimales	boolean

ARTICULOS.DAT (ordenado por **GONDOLA** de mayor a menor)

GONDOLA	CODREPOS	CANTART	PROVEEDOR	CLASE	SUPERVISOR
4 dígitos	4 dígitos	4 dígitos	4 dígitos	1 carácter	3 dígitos

Se pide que imprima un listado ordenado por **GONDOLA** de menor a mayor, respetando el siguiente modelo:

GÓNDOLA	APELLIDO Y NOMBRE DEL REPOSITOR	PREMIOS	CANTIDAD DE ARTÍCULOS
9999	XXXXXXXXXXXXXXXXX	$ 99.99	9999

Nota: *No se dispone de memoria para estructuras estáticas, y la memoria dinámica sólo alcanza si el nodo no supera los 10 bytes.*

43. Se cuenta con el siguiente archivo:

LISTAPRE.DAT (ordenado por **CODIGO_ARTICULO** creciente):

CODIGO_ARTICULO	DESCRIPCION	STOCK	PRECIO
4 dígitos	10 caracteres	2 dígitos	3 dígitos

Se deberá imprimir un listado ordenado por **CODIGO_ARTICULO** decreciente, con todos los artículos en stock, incluyendo todos los datos, y con los artículos faltantes (**STOCK** = 0) se deberá generar un archivo **REPOSIC.DAT** con los mismos registros y orden que **LISTAPRE.DAT**.

44. Se cuenta con los siguientes archivos:

REMISES.DAT (sin orden)

CODAUTO	MARCA	MODELO	KM	DUENIO	SEGURO
4 dígitos	20 caracteres	4 dígitos	7 dígitos	30 caracteres	20 caracteres

CHOFERES.DAT (sin orden)

CODCHOFER	NOMBRE	DNI	CARNETCOND	DIRECCION	TELEFNO	DESDE	HASTA
4 dígitos	25 caracteres	7 dígitos	7 dígitos	20 caracteres	15 caracteres	2 dígitos	2 dígitos

CLIENTES.DAT (sin orden)

CODCLIENTE	NOMBRE	DIRECCION	TELEFONO	SALDO
4 dígitos	20 caracteres	20 caracteres	10 caracteres	2 dígitos y 2 decimales

TARIFAS.DAT (sin orden)

CODDEST	DESTINO	MONTO
4 dígitos	30 caracteres	3 dígitos y 2 decimales

VIAJES.DAT (ordenado por **FECHA**)

FECHA	CODCHOFER	CODAUTO	HORA	TIMPO	CODDEST	CODCLIENTE
AAAAMMDD	4 dígitos	4 dígitos	HHMM	HHMM	4 dígitos	4 dígitos

Se pide que imprima un listado de los viajes realizados, ordenado por fecha decreciente, y respetando el modelo provisto por la agencia:

LISTADO DE VIAJES

FECHA	NOMBRE CHOFER	MARCA AUTO	HORA	TIEMPO DE VIAJE	NOMBRE DESTINO	NOMBRE CLIENTE	MONTO
99/99/99	XXXXXXX	XXXXXXX	HH:MM	HH:MM	XXXXXXX	XXXXXXX	$ 999.99

Nota: *Tener en cuenta que no se dispone de memoria estática y que la memoria dinámica es suficiente*

45. Desarrolle un algoritmo que le pida al usuario la fecha actual, a continuación cargue desde teclado el Nombre, el Apellido y la Fecha de Nacimiento de un grupo de personas, y finalmente imprima en pantalla Nombre y Apellido de cada individuo, seguido de la cantidad de días transcurridos desde su nacimiento, en el mismo orden en que fueron ingresados los datos. El ingreso de datos finaliza con el Nombre FIN.

46. Cree una función **COMPARAR** que cargue de un archivo

VALORES.DAT (sin orden)

VALOR_MEDIDO
5 dígitos y 2 decimales

dos series de mediciones, separadas entre sí por un valor **0**, y que devuelva **TRUE** si ambas series de valores son idénticas, tanto en valor como en posición dentro de la serie, y **FALSE** en caso contrario.

Nota: *No hay memoria para estructura estática y la memoria dinámica es suficiente.*

47. Dadas dos colas **COLA** y **COLB**, ambas del mismo tipo, y ordenadas en forma creciente por el campo **VALOR**, desarrollar y codificar un procedimiento que genere una única cola **COLCONCAT** a partir de ellas, también ordenada por el mismo campo.

48. A partir del siguiente archivo

CUOTAS.DAT (ordenado por **NCUOTA**)

NCUOTA	NCREDITO	MONTO	PAGO
2 dígitos	5 dígitos	2 dígitos y 2 decimales	boolean

Deberá obtenerse un listado de las cuotas impagas, donde se detallará el número de cuota, el número de crédito y el monto de cada cuota, presentando un total al final.

Luego se deberá mostrar un listado de las cuotas pagas, con las mismas características que el de las impagas.

Resuelva el problema para
- *a) 15 bytes por nodo, máximo.*
- *b) 8 bytes por nodo, máximo.*

Nota :*No hay memoria estática disponible.*

49. Recibimos el archivo **FACTURAS.DAT**, el cual se encuentra ordenado por **FECHA** creciente, y cuyo formato de registro es el siguiente:

FECHA	TIPO	NUMERO	CLIENTE	CUIT	CANTART	MONTO	FORMAPAGO
AAAAMMDD	1 carácter	8 dígitos	3 dígitos	10 caracteres	3 dígitos	4 dígitos y 2 decimales	1 carácter

Se nos pide que imprimamos un listado de ventas detallado, discriminado por **TIPO** de factura, el cual puede ser **A, B, C** o **X** para remitos, y dentro de cada tipo, ordenado por **FECHA**.

Resolver el problema considerando las siguientes restricciones:
a) *No hay memoria estática, y la memoria dinámica sólo alcanza si los nodos no superan los 13 bytes.*
b) *Hay 33 bytes de memoria estática, y 204100 bytes de memoria dinámica.*

50. A partir del siguiente archivo histórico de un club social y deportivo:

ACTIVOS.DAT (ordenado por **FECHA** y **DEPORTE**)

FECHA	DEPORTE	CANTSOCIOS	CUOTAADICIONAL	SEDE
AAAAMMDD	3 dígitos	3 dígitos	2 dígitos y 2 decimales	10 caracteres

Se nos solicita que imprimamos:
a) Un listado de inscriptos, discriminado por deporte, y ordenado por **FECHA**
b) Un listado mensual ordenado por fecha con la cantidad total de inscriptos a todos los deportes.

Tener en cuenta que se procesan registros de 20 años diferentes.

Nota: *Hay 1000 bytes de memoria estática. La memoria dinámica es suficiente si los nodos no superan los 14 bytes.*

51. A partir de del siguiente archivo de un videoclub

PELICULA.DAT (ordenada por **TITULO**)

TITULO	DURACION	CALIFICACION	PROTAGONISTA	DIRECTOR	PRODUCTOR	GUIONISTA	MUSICA
20 caracteres	MMM	4 caracteres	30 caracteres	30 caracteres	30 caracteres	30 caracteres	30 caracteres

a) Se pide que se imprima un listado ordenado por **TITULO**, de las películas aptas para todo público (calificadas como 'ATP'), mostrando **TITULO, DURACION** y **PROTAGONISTA**.

No hay memoria estática disponible, y la dinámica sólo alcanza si los nodos no superan los 6 bytes.

b) Se pide un sistema que muestre al usuario la posibilidad de elegir una letra del alfabeto, y a continuación se mostrarán, ordenadas alfabéticamente, todas las películas que comiencen con dicha letra, ordenadas por **TITULO**. Se mostrarán todos los datos de cada película El programa finalizará cuando se teclee 'FIN'.

Hay 220 bytes de memoria estática, y la memoria dinámica es suficiente si los nodos no superan los 6 bytes.

Nota: *Tener en cuenta que no hay más de 45000 películas, y que los títulos se encuentran en mayúsculas.*