

**GUIA DE EJERCICIOS****Ejercicio Nro. 1:**

Dado un conjunto de valores enteros, calcular e imprimir:

- a) cuantos valores cero hubo   b) promedio de los valores positivos   c) sumatoria de valores negativos

Resolver el problema para los siguientes lotes de datos:

- 1) 167 valores enteros.
- 2) N valores, donde el valor de N debe ser leído previamente.
- 3) El conjunto de valores termina con un valor igual al anterior.
- 4) Se dan N valores, pero el proceso deber finalizar si se procesan todos los valores o la cantidad de ceros supera a cuatro.
- 5) Se dan N valores, pero el proceso deberá finalizar si se cumple alguna de las condiciones de 4) o si el promedio de positivos resulta mayor que 6.

**Ejercicio Nro. 2:**

Dado un conjunto de triángulos representados por sus lados L1, L2 y L3 que finaliza con un triángulo con un lado nulo, determinar e imprimir la cantidad de triángulos equiláteros, isósceles y escalenos.

**Ejercicio Nro. 3:**

Dados dos valores N y M determinar e imprimir cuales y cuantos múltiplos de M hay dentro del conjunto 1 a N.

**Ejercicio Nro. 4:**

Dados dos valores enteros A y B determinar e imprimir el producto de ambos obtenido por sumas sucesivas.

**Ejercicio Nro. 5:**

Dado un valor M, determinar e imprimir los M primeros múltiplos de 3 que no lo sean de 5, dentro del conjunto de números naturales.

**Ejercicio Nro. 6:**

Dado un conjunto de valores c/u de los cuales representa el sueldo de un empleado, excepto el último que es cero e indica fin del conjunto.

Desarrollar el programa que determine e imprima:

- a) Cuantos empleados ganan menos de \$520      b) Cuantos ganan \$520 o más pero menos de \$780  
c) Cuantos ganan \$780 o más pero menos de \$999      d) Cuantos ganan \$999 o más

**Ejercicio Nro. 7:**

Dadas 2 circunferencias (X1, Y1, R1) y (X2, Y2, R2) que se intersectan, y un conjunto de puntos (XP, YP) del plano, terminado con uno que coincide con el origen, se debe desarrollar un programa que determine e imprima:

- a) Cantidad de puntos dentro de la circunferencia 1 y fuera de la 2.
- b) Cantidad de puntos dentro de la circunferencia 2 y fuera de la 1.
- c) Cantidad de puntos dentro de la intersección de ambas.
- d) Cantidad de puntos fuera de las dos circunferencias.

Se considera que un punto está dentro de una circunferencia incluso cuando se encuentra sobre la misma.

Ejercicio Nro. 8:

Dado un conjunto de boletos (c/u definido por 5 valores de un dígito) que finaliza con un boleto de 5 ceros, desarrollar un programa que determine e imprima cuantos boletos hubo de los siguientes tipos:

- a) quintuples                      b) cuádruples                      c) triples                      d) dobles

Ejercicio Nro. 9:

Dado un conjunto de ternas de valores (V1, V2, V3) finalizado por una terna nula, desarrollar el programa que imprima con una única orden de impresión los valores de cada terna en orden creciente de izquierda a derecha en la línea.

Ejercicio Nro. 10:

Dado un conjunto de valores positivos terminado con un valor negativo, desarrollar el programa que imprima el valor máximo del conjunto y su posición relativa dentro del mismo.

Ejercicio Nro. 11:

Dado un conjunto de valores finalizado con un valor nulo, desarrollar el programa que determine e imprima (si hubo valores):

- a) Valor máximo negativo    b) Valor mínimo positivo    c) Valor mínimo dentro del rango -17.3 y 26.9

Ejercicio Nro. 12:

En un torneo de fútbol participan K equipos. El torneo se juega con el sistema de todos contra todos. Por cada partido disputado por un equipo se dispone de la siguiente información:

- a) Nro. de equipo                      b) Código del resultado ('P'= Perdido, 'E'= Empatado, 'G'= Ganado)

Se arma un lote de datos con todos los resultados del torneo, agrupados por Nro. de equipo.

Desarrollar el programa que imprima:

- 1) Por cada equipo, su número y el puntaje total que obtuvo (suma 3 si gana, y 1 si empata).
- 2) Nro. de equipo que totalizó la menor cantidad de puntos y cuantos son, si hay más de uno en la misma condición.

Ejercicio Nro. 13:

Se dispone de un conjunto de 8640 temperaturas horarias, que se tomaron durante los 12 meses del año, considerando que todos los meses tienen 30 días y cada día 24 horas. Se debe desarrollar el programa que determine e imprima cual fue, la mayor temperatura y en que mes, día y hora ocurrió.

Ejercicio Nro. 14:

Dado un conjunto de sueldos a pagar (sin centavos), finalizado con un sueldo cero, desarrollar el programa que determine e imprima cuantos billetes de \$100, \$50, \$20, \$10, \$5, \$2 y \$1 son necesarios para ensobrar todos los sueldos y no tener dificultades con el cambio (utilizando la mínima cantidad de billetes).

Ejercicio Nro. 15:

Se dispone de un lote de valores enteros positivos que finaliza con un valor -1. El lote está dividido en sublotes por medio de valores cero.

Desarrollar el programa que determine e imprima:

- a) por cada sublotte el promedio de valores.
- b) el total de sublotes procesados.
- c) Valor máximo del conjunto, indicando el sublotte en que se encontró y la posición relativa del mismo en el sublotte.

Nota: puede estar el lote vacío (primer valor -1), o puede haber uno o varios o todos los sublotes vacíos (ceros consecutivos).

Ejercicio Nro. 16:

Por cada alumno de la facultad se dispone de la siguiente información:

a) Número de legajo                      b) Nota promedio de finales rendidos

Con estos datos se forma un lote terminado con número de legajo cero.

Desarrollar el programa que determine e imprima los números de legajo de los dos mejores alumnos según su promedio.

Ejercicio Nro. 17:

Del censo realizado en una población se conocen los siguientes datos:

a) Día de nacimiento (2 dígitos)

b) Mes de nacimiento (2 dígitos)

c) Año de nacimiento (2 dígitos)

d) Sexo ('M': masculino / 'F': femenino)

Con estos datos de cada habitante se forma un lote finalizado con un día cero. Desarrollar el programa que determine e imprima:

1) Cuantos nacimientos hubo en el mes de octubre de todos los años.

2) Cuantos nacimientos hubo antes del 9 de julio de 1970.

3) Cuantos nacimientos de mujeres hubo en la primavera de 1942.

4) Sexo de la persona más anciana (solo existe una).

Ejercicio Nro. 18:

Desarrollar un programa que dibuje en el monitor un rectángulo centrado:

a) De 12 unidades de alto por 40 unidades de ancho.

b) De A por B. Los valores de A y B deben ser leídos previamente.

Ejercicio Nro. 19:

Desarrollar un programa que dibuje la función seno:

a) En forma horizontal ocupando toda la pantalla    b) En forma vertical ocupando toda la pantalla

Ejercicio Nro. 20:

Se dispone de un conjunto de boletas de inscripción de alumnos a examen en el mes de mayo. Cada boleta tiene los siguientes datos:

a) Nro. de legajo (7 dígitos)    b) Código de materia (6 dígitos)    c) Día del examen (2 dígitos)

d) Mes del examen (2 dígitos)    e) Año del examen (2 dígitos)    f) Nombre y Apellido (25 caracteres)

Desarrollar un programa que a partir de las boletas mencionadas genere un archivo de inscripción de alumnos a exámenes finales.

Al final del proceso imprimir el contenido del archivo para control.

Ejercicio Nro. 21:

Dado el archivo generado en el ejercicio 20, desarrollar un programa que solicitando un código de materia permita seleccionar todos los registros de alumnos que se anotaron para rendirla y los grabe en otro archivo.

Ejercicio Nro. 22:

Dados el archivo del ejercicio 20 que tiene todas las inscripciones del día, y otro con idéntica estructura (registro) que tiene todas las inscripciones anteriores, desarrollar un programa que agregue al archivo anterior el contenido del archivo del día.

Ejercicio Nro. 23:

Modificar el programa del ejercicio 20 para que permita agregar mas inscripciones de alumnos a los exámenes de mayo. Para control sólo imprima los agregados.

Ejercicio Nro. 24:

Dado el archivo del ejercicio 20 ordenado por fecha y para igual fecha por código de materia y para igual materia por número de legajo.

Desarrollar el programa que imprima el listado de alumnos inscriptos en cada materia, según se indica:

	MATERIA: NNNNN	FECHA: NN/NN/NN	
ORDEN	LEGAJO	APELLIDO Y NOMBRES	NOTA
1	NNNNNN-N	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	NN
2	NNNNNN-N	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	NN
.....	NNNNNN-N	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	NN
NNN	NNNNNN-N	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	NN

Ejercicio Nro. 25:

Idem ejercicio anterior pero controlar que la impresión no supere los 50 alumnos por hoja para las materias con inscripciones masivas.

Ejercicio Nro. 26:

Idem ejercicio anterior pero relacionando las hojas para las materias con más de 50 alumnos (ejemplo: Hoja 1/3, Hoja 2/3 y Hoja 3/3 si los alumnos sumaran 135).

Ejercicio Nro. 27:

Dado el archivo MATANT.DAT del ejercicio 20 ordenado por código de materia, resumen de inscripciones de días anteriores, y otro archivo de las mismas características y con el mismo orden MATDIA.DAT con las inscripciones del día de la fecha, desarrollar el programa que aparee los mismos y obtenga el archivo MATACT.DAT resultante con el mismo orden de los anteriores.

Ejercicio Nro. 28:

Dado el archivo MATANT.DAT del ejercicio 20 ordenado por código de materia y fecha, resumen de inscripciones de días anteriores, y otro archivo de las mismas características y con el mismo orden MATFALTA.DAT con las inscripciones faltantes al anterior, desarrollar el programa que aparee (por materia y fecha) los mismos y obtenga el archivo MATACT.DAT resultante con el mismo orden de los anteriores.

Ejercicio Nro. 29:

Dado el archivo ALUMNOS.DAT con los datos personales de alumnos ordenado por legajo, donde cada registro contiene:

- a) Nro. de legajo (7 dígitos)      b) Apellido y nombres (30 caracteres)      c) Domicilio (30 caracteres)  
 d) Código postal (4 dígitos)      e) Teléfono (10 caracteres)      f) Año de ingreso (4 dígitos)
- y otro con el mismo orden que el mencionado llamado NOVEDAD.DAT, con las actualizaciones (altas, bajas y modificaciones) a ser aplicadas, donde cada registro contiene además de todos los campos de ALUMNOS.DAT un código de operación ('A'= Alta, 'B'= Baja, 'M'= Modificación).

Desarrollar el programa que actualice el archivo ALUMNOS.DAT con los registros de NOVEDAD.DAT y obtenga el archivo actualizado ALUMACTU.DAT.

Ejercicio Nro. 30:

Dado el archivo del ejercicio 20 desarrollar un programa que genere un archivo ordenado por número de legajo (cada registro debe tener los campos legajo y apellido y nombres) para todos los alumnos que se inscribieron (una o más veces). Cada legajo debe ocupar una posición única y predecible en el archivo. El rango de legajos es del 80001 a 110000.

Ejercicio Nro. 31:

Se dispone de un archivo ordenado por el campo apellido y nombres, donde cada registro contiene:

- a) Nro. de legajo (1 a 30000, están todos)
- b) Apellido y nombres (30 caracteres)
- c) Total de materias aprobadas (< 99)
- d) Promedio de notas (real)

Desarrollar el programa que genere otro archivo con el mismo tipo y cantidad de registros pero ordenado por legajo.

Nota: se disponen de 64 Kb de memoria.

Ejercicio Nro. 32:

Dado un archivo de alumnos inscriptos para cursar por año completo (desordenado) donde cada registro contiene:

- a) Nro. de legajo (5 dígitos)
- b) Especialidad (1 a 10)
- c) Turno ('M', 'T' o 'N')
- d) División (1 a 100)

Desarrollar el programa que genere otro archivo con la misma cantidad de registros pero ordenado por especialidad, turno y división.

Nota: no hay mas de 100 alumnos por cada división y se disponen de 550 Kb de memoria.

Ejercicio Nro. 33:

Se disponen de dos archivos de registros:

- a) El obtenido como salida del ejercicio 31, con 30000 registros ordenados por legajo.
- b) El obtenido como salida del ejercicio 32.

Desarrollar el programa que imprima el siguiente listado paginado (máximo 50 alumnos por hoja y numeración de hojas Nro./Total en la parte inferior de cada hoja):

ORDEN	ESPECIALIDAD N	LEGAJO	TURNO X	DIVISION 999
			APELLIDO Y NOMBRES	
1		NNNNN	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
2		NNNNN	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
....		NNNNN	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	
50		NNNNN	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	

Nota: Se disponen de 600 Kb de memoria.

Ejercicio Nro. 34:

Dado el archivo ALUMNOS.DAT con los datos personales de alumnos ordenado por legajo, donde cada registro contiene:

- a) Nro. de legajo (6 dígitos)
- b) Apellido y nombres (30 caracteres)
- c) Domicilio (30 caracteres)
- d) Código postal (4 dígitos)
- e) Teléfono (10 caracteres)
- f) Año de ingreso (4 dígitos)
- g) Código de estado ('A'= activo / 'I'= inactivo)

y otro sin ningún orden NOVEDAD.DAT, con los movimientos a ser aplicados, donde cada registro contiene los campos a), b), c), d), e), f) de ALUMNOS.DAT mas un código de operación ('A'= Alta / 'B'= Baja / 'M'= Modificación).

Desarrollar el programa que actualice ALUMNOS.DAT sobre sí mismo en el orden que establecen los registros de NOVEDAD.DAT.

Nota: Se disponen de 600 Kb de memoria, ALUMNOS.DAT tiene un registro por cada legajo (80001 a 110000).

Ejercicio Nro. 35:

Resuelva el ejercicio 34 pero considerando que el archivo ALUMNOS.DAT tiene los 30000 registros pero desordenados.

Nota: El espacio en disco no alcanza para duplicarlo (hay 60 Kb).



Ejercicio Nro. 45:

Ingresar un valor entero  $N$  ( $< 40$ ). A continuación ingresar un conjunto  $A$  y luego otro conjunto  $B$  ambos de  $N$  elementos. Generar un arreglo  $C$  donde cada elemento se forme de la siguiente forma:  
 $C[1] \leftarrow A[1]+B[N]$        $C[2] \leftarrow A[2]+B[N-1]$       .....       $C[N] \leftarrow A[N]+B[1]$

Ejercicio Nro. 46:

Ingresar dos valores enteros  $M$  ( $< 10$ ) y  $N$  ( $< 15$ ). A continuación ingresar un conjunto  $A$  de  $M$  elementos y luego otro  $B$  de  $N$  elementos. Generar e imprimir:

- Un conjunto  $C$  resultante de la anexión de  $A$  y  $B$ .
- Un conjunto  $D$  resultante de la anexión de los elementos distintos de cero de  $A$  y  $B$ .

Ejercicio Nro. 47:

Ingresar dos valores enteros  $M$  ( $< 25$ ) y  $N$  ( $< 10$ ) A continuación ingresar un conjunto  $A$  de  $M$  elementos y luego otro  $B$  de  $N$  elementos, ambos ordenados en forma creciente por magnitud. Generar e imprimir el conjunto  $TOTAL$  resultante del apareo por magnitud de los conjuntos  $A$  y  $B$ .

Ejercicio Nro. 48:

Ingresar un valor entero  $N$  ( $< 40$ ). Luego ingresar un conjunto  $REFER$  de  $N$  elementos reales (ingresan ordenados por magnitud creciente). Finalmente ingresar un valor pesquisa  $X$ . Desarrollar el programa que determine e imprima:

- Con cual elemento (posición) del conjunto coincide, o
- Entre cuales dos elementos (posiciones) se encuentra, o
- Si es menor que el primero o mayor que el último.

Ejercicio Nro. 49:

Ingresar un valor entero  $CANT$  ( $< 50$ ) y a continuación un conjunto  $SINOR$  de  $CANT$  elementos. Desarrollar un programa que determine e imprima:

- El conjunto  $SINOR$  en el que cada elemento original se intercambie por su simétrico:  
 $A[1]$  con  $A[CANT]$ ,  $A[2]$  con  $A[N-1]$ , ..... etc.
- El conjunto  $SINOR$  ordenado de menor a mayor sobre si mismo indicando la posición que ocupaba cada elemento en el conjunto original.

Ejercicio Nro. 50:

Dado un archivo  $PRECIOS$  (desordenado) con los precios de cada articulo, donde cada registro contiene:

- |                                  |   |                             |
|----------------------------------|---|-----------------------------|
| a) Nro. de articulo (5 dígitos)  | b) Descripción del articulo (19 caracteres) | c) Precio por unidad (real) |
| d) Cantidad en stock (5 dígitos) | e) Nro. de proveedor (4 dígitos)            |                             |

Desarrollar el programa que imprima el contenido del archivo ordenado por:

- Nro. de articulo creciente
- Descripción del articulo (alfabético creciente)
- Nro. de proveedor creciente y dentro del mismo por Nro. de articulo creciente
- Nro. de proveedor creciente y dentro del mismo por Nro. de articulo decreciente

Nota: Se dispone de 560 Kb de memoria y hay 20000 artículos como máximo.

Ejercicio Nro. 51:

Una empresa de aviación realiza 500 vuelos semanales a distintos puntos del país y requiere un programa para el otorgamiento de pasajes.

Para ello dispone de un archivo de registros, en el que cada registro contiene información de los vuelos que realiza y la cantidad de pasajes disponibles en cada uno de ellos según se indica:

a) Código del vuelo (6 dígitos) b) Cantidad de pasajes disponibles (3 dígitos)  
Se dispone además de otro archivo con los datos de los potenciales compradores, en el que cada registro tiene:

a) Código de vuelo solicitado b) Cantidad de pasajes solicitados (3 dígitos)  
c) DNI del solicitante (8 dígitos) d) Apellido y nombres del solicitante (30 caracteres)

Desarrollar estrategia, algoritmo y codificación del programa que determine e imprima:

1) Para los solicitantes a los cuales se les venden pasajes,

DNI

APELLIDO Y NOMBRES

CANTIDAD DE  
PASAJES

CÓDIGO DE  
VUELO

99999999

XXXXXXXXXXXXXXXXXXXXXXXXXXXX

999

999999

2) Al final del proceso el siguiente listado:

CÓDIGO DE VUELO

PASAJES LIBRES

PASAJES NO VENDIDOS

999999

999

999

Se le vende al solicitante si la cantidad de pasajes que solicita está disponible, en caso contrario se computa como pasajes no vendidos.

Ejercicio Nro. 52:

Se desarrolla una carrera automovilística de regularidad constituida por 50 trayectos numerados de 1 a 50. Por cada trayecto se generó un registro con el número de trayecto y el tiempo asignado en segundos y se encuentran en el archivo ASIGNADO (sin ningún orden)

a) Nro. del trayecto b) Tiempo asignado en segundos (4 dígitos)

Para llevar el control de los corredores, de posición y de abandonos se dispone de un archivo TIEMPO donde cada registro contiene:

a) Nro. del corredor (3 dígitos) b) Nro. del trayecto c) Tiempo en segundos (4 dígitos)

Los registros de este archivo están ordenados por trayecto pero no por corredor. A partir del abandono de un corredor en un trayecto no habrá más registros para el en el archivo.

Desarrollar estrategia, algoritmo y codificación del programa que determine e imprima:

1) Por cada etapa, su número y el del corredor ganador de la misma.

2) Por cada etapa, su número y los de los corredores que abandonan en la misma.

Ejercicio Nro. 53:

Una empresa que distribuye mercadería hacia distintas localidades del interior dispone de dos archivos de registros:

Uno denominado DESTINOS con información de la distancia a cada uno de los destinos:

a) Nro. de destino (3 dígitos) b) Distancia en kilómetros (NNN.NNN)

Otro denominado VIAJES con los viajes realizados por cada camión (< 200), donde cada registro contiene:

a) Patente del camión (6 caracteres) b) Nro. de destino c) Nro. de chofer (1 a 150)

Desarrollar estrategia, algoritmo y codificación del programa que determine e imprima:

1) Cantidad de viajes realizados a cada destino (solo si > 0).

2) Nro. de chofer con menor cantidad de Km (entre los que viajaron).

3) Patente de los camiones que viajaron al destino 116 sin repeticiones de las mismas.

Ejercicio Nro. 54:

Ingresar dos valores, M (< 30) y N (< 25) y a continuación por filas todos los componentes de una matriz MATRIZA de M filas y N columnas. Desarrollar un programa que:

a) Imprima la matriz MATRIZA por columnas.

b) Calcule e imprima el valor promedio de los componentes de la matriz.

c) Genere e imprima un vector VECSUMCOL donde cada componente sea la suma de la columna homóloga.

d) Genere e imprima un vector VECMAXFIL donde cada componente sea el valor máximo de cada fila.



Ejercicio Nro. 55:

Ingresar un valor  $N$  ( $< 25$ ) y luego por filas una matriz cuadrada CUADRA de  $N$  filas y columnas. Desarrollar un programa que determine e imprima:

- Todos los elementos de la diagonal principal o secundaria según de cual resulte mayor la sumatoria de elementos.
- Los elementos del cuarto ( $N/2$  filas y  $N/2$  columnas) cuya sumatoria resulte mayor (considerando que  $N$  fuera par).
- Los elementos de la triangular superior o inferior dependiendo de cual tenga mayor sumatoria de elementos.

Ejercicio Nro. 56:

Se da un conjunto de 744 temperaturas que se tomaron en una localidad diariamente hora por hora durante los 31 días de un mes (el conjunto esta ordenado por día y hora). Desarrollar un programa que determine e imprima:

- Armar la matriz TEMPEDIA en que cada fila corresponda a un día y cada columna a una hora e imprimirla por horas.
- En que día y hora se produjo la temperatura máxima del mes.
- En que día se produjo la menor temperatura media.
- A que hora se produjo la mayor temperatura media.

Ejercicio Nro. 57:

Ingresar dos valores,  $M$  ( $< 20$ ) y  $N$  ( $< 25$ ) y a continuación por columnas todos los componentes de una matriz DESORDE de  $M$  filas y  $N$  columnas. Desarrollar un programa que:

- Ordene (creciente) cada columna de la matriz sobre si misma y la imprima a razón de una columna por renglón.
- Ordene (creciente) la matriz sobre si misma por fila desde el elemento 1,1 al  $M,N$  y la imprima a razón de una fila por renglón.

Ejercicio Nro. 58:

Ingresar por plano, fila y columna todos los elementos de una matriz MATRIDIM de  $M$  planos, filas y columnas. Desarrollar un programa que:

- Imprima la matriz MATRIDIM por columnas, fila, plano.
- Calcule e imprima el valor promedio de la matriz.
- Determine e imprima el mayor valor y en que lugar de la matriz se encuentra.
- Genere e imprima una matriz MATCSUMCOL donde cada elemento sea la suma de la columna homóloga.
- Genere e imprima una matriz MATMAXFIL donde cada elemento sea el valor máximo de cada fila.

Ejercicio Nro. 59:

Dada una pila y un valor  $I$ , desarrollar un procedimiento que elimine los 2 primeros nodos de la pila y deje el valor  $I$  como primero. (Definir parámetros y codificar).

Ejercicio Nro. 60:

Dada una pila y un valor  $I$ , desarrollar un procedimiento que inserte  $I$  como tercer valor de la pila. (Definir parámetros y codificar).

Ejercicio Nro. 61:

Idem ejercicio 60 pero retornando un parámetro con valor 'S' o 'N' según haya sido exitoso o no el requerimiento. (Definir parámetros y codificar).

Ejercicio Nro. 62:

Dada una pila y dos valores X e I, desarrollar un procedimiento que inserte el valor X en la posición I de la pila si es posible. (Definir parámetros y codificar).

Ejercicio Nro. 63:

Dada una pila y un valor X, desarrollar un procedimiento que inserte el valor X en la última posición de la pila y la retorne. (Definir parámetros y codificar).

Ejercicio Nro. 64:

Dada una pila y dos valores X e Y, desarrollar un procedimiento que reemplace cada valor igual a X que se encuentre en la pila por el valor Y retornando la pila modificada. En caso de no haber ningún valor igual a X retornar la pila sin cambio. (Definir parámetros y codificar).

Ejercicio Nro. 65:

Definir una función INVERSA que evalúe dos conjuntos de caracteres separados por un punto y retorne True si los conjuntos son inversos (ej: ABcDe.eDcBA) o False si no lo son. Los conjuntos deben ingresarse por teclado. (Definir parámetros y codificar).

Ejercicio Nro. 66:

Desarrollar un procedimiento que ingrese por teclado un conjunto de Apellidos y Nombre de alumnos y los imprima en orden inverso al de ingreso. (Definir parámetros y codificar).

Ejercicio Nro. 67:

Dada una pila desarrollar un procedimiento que ordene la misma de acuerdo al valor de sus nodos y la retorne. Solo se deben usar pilas. (Definir parámetros y codificar).

Ejercicio Nro. 68:

Dada una cola (nodo = registro + puntero), desarrollar y codificar un procedimiento que elimine 4 nodos de la misma (indicar con un parámetro 'S'/'N' si ello fue, o no posible)

Ejercicio Nro. 69:

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

Ejercicio Nro. 70:

Dadas dos colas COLA y COLB (nodo = registro + puntero), desarrollar y codificar un procedimiento que genere una única cola COLAB a partir de ellas. (Primero los nodos de COLA y luego los de COLB).

Ejercicio Nro. 71:

Dada una cola (nodo = registro + puntero), imprimirla en orden natural si tiene más de 100 nodos, caso contrario imprimirla en orden inverso.

Ejercicio Nro. 72:

Dadas dos colas COLA y COLB (nodo = registro + puntero), desarrollar y codificar un procedimiento que genere otra cola COLAB por apareo del campo ARRIBO del registro (define orden creciente en ambas).

Nota: COLA y COLB dejan de ser útiles después del apareo.

Ejercicio Nro. 73:

Dado un archivo de registros de alumnos, donde cada registro contiene:

- a) Apellido y Nombre del alumno (35 caracteres)
- b) Número de legajo (7 dígitos)
- c) División asignada (1 a 100)

ordenado por número de legajo, desarrollar el algoritmo y codificación del programa que imprima el listado de alumnos por división, ordenado por división y número de legajo crecientes, a razón de 55 alumnos por hoja.

Ejercicio Nro. 74:

Idem Ejercicio Nro. 73, pero el listado de alumnos por división debe realizarse ordenado creciente por división y decreciente por número de legajo.

Ejercicio Nro. 75:

Idem Ejercicio Nro. 73 pero considerando que las divisiones asignadas son 100 y se identifican con un código de 4 caracteres.

Ejercicio Nro. 76:

Dado un arreglo de N ( $< 30$ ) colas (nodo = registro + puntero), desarrollar y codificar un procedimiento que aparee las colas del arreglo en las mismas condiciones que las definidas en el Ejercicio Nro. 72.

Nota: Retornar la cola resultante y no mantener las anteriores.

Ejercicio Nro. 77:

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

Ejercicio Nro. 78:

Dadas dos listas LISTA y LISTB (nodo = registro + puntero), desarrollar y codificar un procedimiento que genere una única lista LISTC a partir de ellas. (Primero los nodos de LISTA y luego los de LISTB).

Ejercicio Nro. 79:

Dada una LISTA (nodo = registro + puntero), imprimirla en orden natural si tiene más de 100 nodos, caso contrario imprimiría en orden inverso.

Ejercicio Nro. 80:

Dadas dos listas LISTA y LISTB (nodo = registro + puntero), desarrollar y codificar un procedimiento que genere otra lista LISTC por apareo del campo LEGAJOS del registro (define orden creciente en ambas).

Nota: LISTA y LISTB dejan de ser útiles después del apareo).

Ejercicio Nro. 81:

Dado un archivo de registros de alumnos ARCHA sin ningún orden, donde cada registro contiene:

- a) Apellido y Nombre del alumno (34 caracteres)
- b) Número de legajo (6 dígitos)
- c) División asignada (1 a 100)

Se debe desarrollar el algoritmo y codificación del programa que genere un archivo ARCHL igual al anterior pero ordenado por número de legajo.

Nota: Memoria estática 64 Kb; dinámica suficiente si ningún nodo ocupa más de 12 bytes.

Ejercicio Nro. 82:

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

- a) Apellido y Nombre del alumno (34 caracteres)
- b) Número de legajo (7 dígitos)
- c) División asignada (3 dígitos)

Se debe desarrollar el algoritmo y codificación del programa que imprima el listado de alumnos por división ordenado por división y número de legajo crecientes, a razón de 55 alumnos por hoja.

Considerar las siguientes alternativas:

- 1) Memoria estática 4 Kb; dinámica suficiente si ningún nodo ocupa más de 12 bytes
- 2) Memoria estática 3 Kb; dinámica suficiente si ningún nodo ocupa más de 14 bytes
- 3) Memoria estática 3 Kb; dinámica suficiente si ningún nodo ocupa más de 12 bytes

Ejercicio Nro. 83:

Igual planteo que Ejercicio Nro. 82, pero con dos archivos de alumnos sin orden ARCHA y ARCHB.

Considerar las siguientes alternativas:

- 1) Memoria estática 8 Kb; dinámica suficiente si ningún nodo ocupa más de 12 bytes
- 2) Memoria estática 1 Kb; dinámica suficiente si ningún nodo ocupa más de 15 bytes
- 3) Memoria estática 2 Kb; dinámica suficiente si ningún nodo ocupa más de 13 bytes
- 4) Memoria estática 3 Kb; dinámica suficiente si ningún nodo ocupa más de 12 bytes

## EXÁMENES FINALES

20/02/97

Una facultad requiere una estadística con relación a los exámenes finales tomados en la misma. Para ello dispone de un archivo de exámenes FINALES ordenado por fecha, donde cada registro contiene:

- |   |                                     |
|---|-------------------------------------|
| a) Número. de legajo (7 dígitos)        | b) Código de materia (6 caracteres) |
| c) Código. de especialidad (1 carácter) | d) Número. de libro (3 dígitos)     |
| e) Número de folio (1 a 200)            | f) Número de orden (1 a 20)         |
| g) Nota (0 a 10)                        | h) Fecha (AAMMDD)                   |

Dispone además de un archivo con los datos personales de alumnos DATPER, ordenado por legajo, donde cada registro contiene:

- |                              |                                       |
|------------------------------|---------------------------------------|
| a) Número de legajo          | b) Apellido y Nombres (30 caracteres) |
| c) Domicilio (40 caracteres) | e) Datos varios (43 caracteres)       |

Desarrollar la metodología necesaria para obtener un programa que:

- 1) Imprima la información del archivo FINALES por libro, folio y orden según se indica:

LISTADO DE LIBROS DE EXAMENES				
	LIBRO 999	FOLIO 999	FECHA DD/MM/AA	
ORDEN	LEGAJO	CODIGO DE MATERIA	APPELLIDO Y NOMBRES	NOTA ESPECIA
1	9999999	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	5 CINCO X
2	9999999	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	6 SEIS X
	RINDIERON 99	APROBARON 99	DESAPROBARON 99	

- 2) Imprima un listado de exámenes rendidos por cada alumno, ordenado por fecha, con el siguiente formato:

LISTADO DE EXAMENES POR ALUMNO					
LEGAJO: 9999999	APPELLIDO Y NOMBRES: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX				
	MATERIA	LIBRO	FOLIO	FECHA	NOTA
	999999	999	999	99/99/99	3 TRES
	999999	999	999	99/99/99	6 SEIS

Nota: No hay alumnos que rindan mas de 160 exámenes. Para estructuras dinámicas la memoria es suficiente.

28/02/97

Un organismo provincial debe imprimir las patentes de su parque automotor, y para ello cuenta con los siguientes archivos de registros:

De vehículos VEHICULO, (ordenado por código postal y patente) donde c/registro contiene:

- |                            |                                   |                                  |
|----------------------------|-----------------------------------|----------------------------------|
| a) Modelo (4 dígitos)      | b) Año de fabricación (2 dígitos) | c) Número de patente (8 dígitos) |
| d) Titular (35 caracteres) | e) Domicilio (30 caracteres)      | f) Código postal (4 dígitos)     |

De IMPORTES (ordenado por modelo y año de fabricación) donde cada registro contiene:

- |           |                       |                             |
|-----------|-----------------------|-----------------------------|
| a) Modelo | b) Año de fabricación | c) Importe anual (99999.99) |
|-----------|-----------------------|-----------------------------|

De infracciones INFRACCI, (ordenado por patente) donde cada registro contiene:

- |                      |                                       |
|----------------------|---------------------------------------|
| a) Número de patente | b) Cantidad de infracciones (0 a 999) |
|----------------------|---------------------------------------|

Hay un registro por cada vehículo.

Desarrollar un programa que:

- 1) Pida la cantidad de cuotas del año ( $\leq 12$ ) y sus fechas de vencimiento (DDMMAA)

- 2) Imprima para cada vehículo sus cuotas con el siguiente formato:

PATENTE: 99999999	TITULAR: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		
COD.POS.: 9999	DOMICILIO: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX		CANT.INFR.: 999
CUOTA	IMPORTE	FECHA DE VENCIMIENTO	
1	9999.99	DD/MM/AA	
2	9999.99	DD/MM/AA	

Ordenado por número de patente.

- 3) Para los vehículos con mas de 50 infracciones genere un archivo de cuotas a cobrar con todas las cuotas generadas para las mismas, ordenado por número de cuota y patente, donde cada registro contenga:

- |           |    |                    |            |                                  |
|-----------|----|--------------------|------------|----------------------------------|
| a) Número | de | b) Número de cuota | c) Importe | d) Fecha de vencimiento (DDMMAA) |
|-----------|----|--------------------|------------|----------------------------------|

patente

Nota: Nota utilizar procedimientos y funciones. Utilizar estructuras dinámicas con punteros (para el punto 3).

05/03/97

Un ferrocarril desea realizar las ventas de pasajes de sus trenes. Cada tren está constituido por una formación de 10 vagones con 80 asientos cada uno, distribuidos de la siguiente forma:

Vagones 1/3 = Categoría 1      Vagones 4/8 = Categoría 2      Vagones 9/10 = Categoría 3

Para ello se dispone de un archivo de registros TRENES ordenado por código de tren y día, con la información del estado de disponibilidad de cada vagón, de cada tren, cada uno de los cuales con el siguiente formato:

a) Código de tren (15 caracteres)      b) Día (1 a 31)      c) Destino (30 caracteres)

y por cada de los 10 vagones:

d.1) Vagón (1 a 10)      d.2) Cantidad de asientos vendidos (0 a 80)

Luego se debe procesar un archivo de registros PASAJES cronológico, con los pedidos de pasajes de cada usuario, donde cada registro contiene:

a) Código de tren      b) Día      c) Categoría

d) Pedido de pasajes (1 a 80)      e) Número de documento (8 dígitos)

Desarrollar la metodología necesaria para obtener el programa que:

1) Imprima por cada venta realizada:

DOCUMENTO	DIA	CODIGO DE TREN	CATEGORIA	VAGON	NUMEROS DE ASIENTOS
99999999	99	XXXXXXXXXXXXXXX	9	99	99 a 99
99999999	99		9	99	99 a 99

2) Imprima al final del proceso (sin repeticiones del solicitante):

#### LISTA DE ESPERA

CODIGO DE TREN	DIA	CATEGORIA	CANTIDAD	DOCUMENTO
XXXXXXXXXXXXXXX	99	9	99	99999999
XXXXXXXXXXXXXXX	99	9	99	99999999

Nota: Todos los pasajes del pedido deben estar en el mismo vagón para ser satisfecho, sino el pedido pasará a lista de espera. Como máximo hay 1000 trenes y se disponen de 64 Kb de memoria estática. El punto 2) debe ser resuelto con estructuras dinámicas.

23/05/97

Dado un archivo ordenado por especialidad y división con la cantidad de vacantes de cada aula de la facultad, donde cada registro contiene:

a) Número de especialidad (2 dígitos)      b) Número de división (1 a 40 TM, 41 a 60 TT, 61 a 99 TN)

c) Cantidad de vacantes (2 dígitos)      d) Cantidad de inscriptos (2 dígitos, inicialmente en cero)

Se dispone además de un archivo en orden cronológico de los alumnos de primer año que solicitan inscribirse, donde c/registro contiene:

a) Número de legajo (6 dígitos)      b) Apellido y nombres (30 caracteres)  
c) Número de especialidad      d) Código de turno (1 carácter: M, T o N)

Desarrollar los pasos necesarios para obtener el programa que:

1) Asigne división a c/solicitante y actualice el archivo, grabando además un registro con todos los datos del alumno y además el número de división o cero en caso de no ser posible la asignación.

2) Haga un listado cronológico de los solicitantes de cada turno de cada especialidad, utilizando estructuras dinámicas, según se indica:

NUMERO DE ORDEN ORIGINAL	ESPECIALIDAD 9	LEGajo	TURNOS X	APELLIDO Y NOMBRES
99999		999999		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
99999		999999		XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

Nota: Se dispone de 1 Mb de memoria.

23/07/97

Una compañía de seguros requiere controlar los pagos mensuales de las pólizas de sus clientes. Para ello cuenta con:



e) Estado (A adeudada / C cancelada) f) Fecha de cobro (AAMMDD)  
DATOPERS (menos de 8000 registros ordenados por código de contribuyente) donde cada registro contiene:

- a) Código de contribuyente b) Apellido y nombres (30 caracteres)  
c) Domicilio (46 caracteres) d) Código de zona municipal (4 caracteres)

COBROMES (ordenado por fecha de cobro y código de contribuyente) donde cada registro contiene:

- a) Código de contribuyente b) Año de la cuota  
c) Mes de la cuota d) Fecha de cobro (AAMMDD)

Desarrollar la estrategia, algoritmo y codificación del programa que:

- 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 PAGOS

CONTRIB	APELLIDO Y NOMBRES	AÑ O	ME S	IMPORT E	P	AÑ O	MES	IMPORT E	P
XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	99	99	9999.99	P	99	99	9999.99	E
		99	99	9999.99	E	99	99	9999.99	E
XXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	99	99	9999.99	P	99	99	9999.99	P

Indique el resultado con P: pago aceptado / E: cuota pagada anteriormente

- 3) Imprima el total de lo recaudado por cada zona municipal ordenado por código de zona:

CODIGO DE ZONA	IMPORTE RECAUDADO
XXXX	9999999.99
XXXX	9999999.99

Nota : Se disponen de 4 Mb de memoria y 4 MB en disco.

02/12/97

Una empresa requiere información de las ventas de un mes de sus sucursales. Para ello cuenta con:

Un archivo CLIENTES de clientes ordenado por código de cliente en el que cada registro contiene:

- a) Código de cliente (6 caracteres) b) Razón Social (30 caracteres) c) Domicilio (45 caracteres)

Un archivo de ventas VENTMES (ordenado por día y sucursal) en el que cada registro contiene:

- a) Número de factura (8 dígitos) b) Fecha (DDMMAA) c) Importe (999999.99)  
d) Código de cliente e) Código de sucursal (4 dígitos)

Desarrollar la estrategia, algoritmo y codificación del programa que imprima:

- 1) Un listado del total de venta a cada cliente ordenado por código de cliente (solo los que compraron en 13 o más días del mes)

TOTAL DE VENTAS MENSUALES A CLIENTES

CODIGO CLIENTE	DOMICILIO	IMPORTE
XXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX	999999.99

- 2) Un listado de la facturación de c/sucursal, con totales por día y sucursal según se indica:

NÚMERO DE FACTURA	SUCURSAL 9999	DIA 99	IMPORTE
99999999	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		999999.99
99999999	XXXXXXXXXXXXXXXXXXXXXXXXXXXX		999999.99
	TOTAL DEL DÍA	999999.99	
	TOTAL DE LA SUCURSAL	99999999.99	

Nota: Cantidad máxima de memoria estática = 100 Kb. El tamaño máximo de cada nodo es de 13 bytes.

10/12/97

Una empresa de seguros para automotores requiere la siguiente información:

- 1) Se actualice la deuda de cada vehículo y se obtenga un listado de los que están al día con las cuotas (1 a 12) pagadas, ordenado por número de legajo según se indica:

LISTADO DE PAGOS EN MESES 1 A 99

PATENTE	NOMBRE DEL PROPIETARIO	1	2	3	4	5	6	7	8	9	10	11	12
XXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	1	3	3	3	5	7	7	9	9			



XXXXXX XXXXXXXXXXXXXXXXXXXX

4 4 7 7 7 9 9

- 2) Un listado por zona geográfica de los vehículos que se encuentran sin cobertura, (adeudan una o más cuotas), con el siguiente formato:

PATENTE	NOMBRE DEL PROPIETARIO	ZONA XXXX	MESES QUE ADEUDA
XXXXXX	XXXXXXXXXXXXXXXXXXXXX	2 4 5 9	
XXXXXX	XXXXXXXXXXXXXXXXXXXXX	3 6 7	

Al final de cada zona imprima la cantidad total de cuotas adeudadas.

Para obtener esta información la empresa dispone estos archivos:

De vehículos VEHIC.DAT, donde cada registro contiene:

- |   |                               |
|---|-------------------------------|
| a) Patente (6 caracteres)                 | b) Código de zona (3 dígitos) |
| c) Nombre del propietario (25 caracteres) | d) Fecha de alta (DDMMAA)     |

De pagos del año de cada vehículo PAGANOS.DAT, donde cada registro tiene:

- |            |   |                               |
|------------|---|-------------------------------|
| a) Patente | b) Meses pagados (12 campos byte, que tienen el mes en que se pagó o 0) | c) Importe total de los pagos |
|------------|---|-------------------------------|

De pagos del mes PAGMES.DAT, donde cada registro contiene:

- |            |                           |                   |
|------------|---------------------------|-------------------|
| a) Patente | b) Mes pagado (2 dígitos) | c) Importe pagado |
|------------|---------------------------|-------------------|

Nota: Ingresar el mes de referencia (proceso). Ninguno de los archivos está ordenado. El tamaño máximo de cada nodo es de 18 bytes. Los dos primeros archivos tienen la misma cantidad de registro.

18/12/97

Una empresa desea realizar estadísticas en función de sus ventas a lo largo de un año, para ello cuenta con los siguientes archivos:

De facturas a clientes FACTAÑO donde cada registro tiene:

- |                   |                                    |                                 |
|-------------------|------------------------------------|---------------------------------|
| a) Fecha (DDMMAA) | b) Número de cliente (999999)      | c) Número de factura (99999999) |
| d) Items (1 a 20) | d.1.1) Número de artículo (99999)  | d.1.2) Importe (99999.99)       |
| .....             | d.20.1) Número de artículo (99999) | d.20.1) Importe (99999.99)      |

Ordenado por zona, sucursal y fecha (20 ítems. Los no utilizados contienen cero en los dos campos).

De CLIENTES sin orden, donde cada registro contiene:

- |                        |                                      |                              |
|------------------------|--------------------------------------|------------------------------|
| a) Número de cliente   | b) Nombre y apellido (30 caracteres) | c) Domicilio (45 caracteres) |
| d) Zona (4 caracteres) | e) Código postal (9999)              | f) Sucursal (999)            |

Desarrolle la estrategia, algoritmo y codificación del programa que:

- 1) Imprima un listado por cada sucursal, con sus 50 mejores clientes (los que más compraron en la sucursal), según se indica:

NÚMERO DE CLIENTE	SUCURSAL 999	NOMBRE Y APELLIDO	IMPORTE TOTAL
999999		XXXXXXXXXXXXXXXXXXXXXXXXXXXXX	999999.99
999999		XXXXXXXXXXXXXXXXXXXXXXXXXXXXX	999999.99

No listar las sucursales sin clientes con compras.

- 2) Genere un archivo ordenado por número de artículo donde cada registro contenga:

- |                       |   |
|-----------------------|---|
| a) Número de artículo | b) 12 campos, cada uno de ellos con el importe total vendido en cada mes (al menos uno de ellos mayor que cero) |
|-----------------------|---|

Nota: Máximo tamaño de nodo 20 bytes. Espacio libre en disco 16 Mb. Memoria estática 16 Kb. Memoria dinámica suficiente.

17/02/98

Una empresa de telefonía móvil, requiere facturar las comunicaciones de todos sus usuarios. Para ello dispone de los siguientes archivos:

De USUARIOS (ordenado por número de usuario), en el que cada registro contiene:

- |  |                                 |
|--|---------------------------------|
| a) Número de usuario (999999999)         | b) Razón Social (36 caracteres) |
| c) Domicilio y localidad (53 caracteres) | d) Categoría (99)               |

De CATEGORIA (sin orden), en el que cada registro contiene:

- |              |  |
|--------------|--|
| a) Categoría | b) Cantidad de pulsos sin cargo (9999) |
|--------------|--|

c) Importe del abono (999.99) d) Importe por cada pulso excedido (9.99)  
De comunicaciones COMUNICA (ordenado por número de usuario), en el que cada registro contiene:

- a) Número de usuario b) Cantidad de pulsos (9999)  
c) Día del mes (1 a 31) d) Hora y minutos de inicio (HHMM)

Desarrollar la metodología necesaria para obtener un programa que:

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

RAZÓN SOCIAL: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX NÚMERO DE USUARIO: 999999999

DOMICILIO: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX

ABONO (9999 PULSOS SIN CARGO)	999.99
ADICIONAL (9999 x 9.99)	9999.99
TOTAL	9999.99

- 2) Imprima al final una lista (ordenada por categoría y usuario) de los usuarios que tuvieron mas de 50 comunicaciones en alguno de los días:

LISTADO DE COMUNICACIONES DIARIAS - CATEGORÍA 99												
NÚMERO DE USUARIO	1	2	3	..	..	..	..	..	..	29	30	31
999999999	999	999	999	..	..	..	..	..	..	999	999	999
999999999	999	999	999	..	..	..	..	..	..	999	999	999

Nota: La memoria disponible es suficiente para estructuras dinámicas si se utiliza un nodo por usuario que cumpla con lo indicado en 2). Utilizar procedimientos y funciones con parámetros.

24/02/98

Una compañía de seguros de automotores requiere información acerca de los siniestros pagados a sus clientes. Para ello cuenta con:

Un archivo POLIZAS.DAT, en el que cada registro contiene:

- a) Código de póliza (8 caracteres) b) Importe original (real) c) Apellido y nombres (30 caracteres)  
d) Domicilio (45 caracteres) e) Importe ajustado (real) f) Año de origen (2 dígitos)

Ordenado por código de póliza.

Un archivo SINIESTR.DAT con los siniestros de los años 93 a 97, donde cada registro contiene:

- a) Código de póliza b) Fecha (numérico AAMMDD)  
c) Código del siniestro (3 dígitos) d) Importe pagado (real)

Ordenado por fecha creciente.

Desarrollar estrategia, algoritmo y programa que realice:

- 1) Listado de siniestros con orden código de póliza y fecha decreciente.

APELLIDO Y NOMBRES: XXXXXXXXXXXXXXXXXXXXXXXXXXXX CÓDIGO DE PÓLIZA: XXXXXXXX

ORDEN DEL SINIESTRO	FECHA	CÓDIGO DEL SINIESTRO	IMPORTE PAGADO
1	DD/MM/AA	999	999999.99
2	DD/MM/AA	999	999999.99

- 2) Actualice la prima en el archivo POLIZAS.DAT si corresponde de acuerdo al siguiente criterio:

Descuento 2% por año sin siniestro, desde el 97 hacia el 93 (mínimo 0% - máximo 10%).

Nota: Tamaño máximo del nodo 8 bytes. Espacio en disco Fsize(POLIZAS) \* 4 bytes

3/03/98

La Ciudad de Buenos Aires requiere un estudio estadístico de infracciones realizadas por todo tipo de vehículo en sus calles.

Para ello cuenta con 2 archivos INFZONA1 e INFZONA2, cada uno de ellos con infracciones de los 12 meses del año 1997, correspondientes a las 2 zonas en que se encuentra dividida la ciudad, para los que cada registro contiene:

- a) Código de patente (8 caracteres) b) Número de registro del conductor (8 dígitos)  
c) Fecha de la infracción (AAMMDD) d) Zona (6 caracteres)  
e) Código de la infracción (5 dígitos) f) Placa del policía (6 caracteres)

Ambos ordenados por zona y fecha creciente.

Desarrollar la metodología necesaria para obtener un programa que:

- 1) Genere un único archivo INFTOTAL 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.



5/08/98

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

- a) De conexiones CONEXION (orden cronológico), en el que cada registro contiene:
  - a.1) Día del mes (1 a 31)
  - a.2) Número de usuario (8 dígitos)
  - a.3) Código de destino (16 caracteres)
  - a.4) Cantidad de minutos (3 dígitos)
- b) De categoría CATEGORI (sin orden), en el que cada registro contiene:
  - b.1) Categoría (2 dígitos)
  - b.2) Minutos sin cargo adicional (4 dígitos)
  - b.3) Importe del abono (999.99)
  - b.4) Importe por minuto adicional (9.99)
- c) De usuarios USUARIO (ordenado por número de usuario), en el que cada registro contiene:
  - c.1) Número de usuario
  - c.2) Apellido y nombres (34 caracteres)
  - c.3) Domicilio (49 caracteres)
  - c.4) Categoría

Desarrollar la estrategia, algoritmo y codificación del programa que:

- 1) Imprima la factura de los usuarios (ordenadas por número de usuario) según se indica:

USUARIO N° 99999999 APELLIDO Y NOMBRES: XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
DOMICILIO: XX

ABONO (999 minutos sin cargo)	999.99
ADICIONAL (999 x b.4)	999.99
TOTAL	9999.99

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

CODIGO DE DESTINO: XXXXXXXXXXXXXXXX	APELLIDO Y NOMBRES	MINUTOS
XXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	9999
XXXXXXXXXXXXXXXXXXXXXXXXXXXX	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	9999

Nota: Hay 126.5 Kb de memoria estática. Nodo máximo = 8 bytes. No hay espacio en disco. Cantidad máxima de destinos 5.000.

22/09/98

Una empresa de turismo representante de varias compañías aéreas, para atender las demandas de viajes para un mes, reserva con anticipación una cantidad de pasajes aéreos a diferentes ciudades del mundo (solo un vuelo por ciudad y día). Esta información se encuentra en el archivo RESEPASA.DAT (sin orden), donde cada registro contiene:

- a) Código de ciudad (999)
- b) Precio de cada pasaje (9999.99)
- c) Día de salida (99)
- d) Hora de salida (HHMM)
- e) Número de vuelo (9999)
- f) Total pasajes reservados (999)

Por otro lado se dispone de un archivo PASOLICI.DAT (sin orden) con los pasajes solicitados por sus clientes, donde cada registro contiene:

- a) Código de ciudad
- b) Cantidad solicitada (999)
- c) Día de salida
- d) Importe a pagar (0)
- e) Número de documento (99999999)

Desarrollar la estrategia, algoritmo y codificación del programa que:

- 1) Imprima para el día en que se rechazaron más pedidos, un listado de rechazos (ordenado por Código de Ciudad creciente) como se indica:

DIA DE MAXIMO RECHAZO 99		
NUMERO DE DOCUMENTO	CODIGO DE CIUDAD	CANTIDAD SOLICITADA
99999999	999	999

- 2) Actualice el campo Importe a Pagar del archivo PASOLICI.DAT y el campo total pasajes reservados de RESEPASA.DAT cuando corresponda.

- 3) Compactar el archivo PASOLICI.DAT grabando todos los registros aceptados desde el inicio y terminados con un registro con código de ciudad cero.

Nota: El programa debe usar memoria estática < 316.000 bytes, nodos de 8 bytes como máximo y no usar archivos adicionales.

11/12/98

Un club deportivo tiene un archivo CAT con los datos por cada una de las 36 categorías de socios:

a) Categoría (6 caracteres)

b) Sexo (M= Mujer / V= varón)

c) Edad mínima en años

d) Edad máxima en años

Se dispone además de un archivo CENSO con los datos obtenidos de cada socio con sus deportes preferidos, cada registro con:

a) Número de socio (6 dígitos)

b) Edad en años

c) Sexo

d) 25 campos tipo Char correspondientes a los deportes 1 a 25 con X o blanco según sea preferido o no

Ordenados por número de socio, pudiendo haber más de un registro por socio, pero solo se debe considerar al socio una vez por deporte preferido.

Se debe desarrollar la estrategia, algoritmo y codificación en Pascal del programa que imprima:

1) El siguiente cuadro de relación Deporte/Categoría:

CATEGORIA					DEPORTES PRACTICABLES				
1	2	3	4		22	23	24	25	
XXXXXX	SI	SI	NO	SI	.....	NO	NO	NO	SI
XXXXXX	NO	SI	SI	NO	.....	NO	SI	NO	NO

Se considera deporte practicable para una categoría si se logran más de 50 socios interesados en la misma.

2) Para los deportes practicables, un listado de los socios interesados ordenado por deporte, categoría y número de socio.

Nota: Archivo en disco 0 bytes. Tamaño máximo de nodo = 8 bytes. Memoria estática = 10 Kb. Usar procedimientos y funciones.

18/12/98

Una farmacia requiere un mejor control de la reposición de medicamentos por parte de las droguerías que le proveen.

Para ello cuenta con un archivo STOCK (ordenado por medicamento) con la siguiente información por cada uno de los medicamentos que comercializa:

a) Número de medicamento (8 dígitos)

b) Cantidad disponible (4 dígitos)

c) Cantidad mínima (4 dígitos)

d) Código de droguería (5 caracteres)

e) Fecha de última venta (AAMMDD)

f) Importe de venta por unidad (999.99)

Además se tiene otro archivo VENTAM con un registro por medicamento vendido en el mes (ordenado por fecha) con la siguiente información:

a) Número de medicamento b) Cantidad de unidades (5 dígitos) c) Fecha de venta (AAMMDD)

Desarrollar la estrategia, algoritmo y codificación del programa que:

1) Actualice archivo STOCK (cada medicamento sólo una vez por día).

2) Realice un listado por día de los medicamentos vendidos en el mismo, ordenado (creciente) por importe diario de venta de cada medicamento.

DIA 99	NUMERO DE MEDICAMENTO	IMPORTE TOTAL
	99999999	999999.99
	99999999	999999.99

DIA 99	NUMERO DE MEDICAMENTO	IMPORTE TOTAL
	99999999	999999.99

3) Imprima un listado de los medicamentos que no tuvieron movimiento en el mes, ordenado por droguería, según se indica:

CODIGO DE DROGUERIA XXXXX	MEDICAMENTOS SIN MOVIMIENTO
	99999999
	99999999

Nota: Memoria estática = 0 bytes. Nodo máximo = 14 bytes.

Cantidad máxima de nodos por medicamento = 1. Bytes disponibles para archivo = 0

29/12/98

Un supermercado requiere procesar los tickets de su venta diaria y obtener cierta información de la misma.

Para ello cuenta con un archivo ARTIC con un registro por artículo cada uno de los cuales contiene:

a) Código de artículo (7 caracteres)

b) Precio por unidad (real)

c) Datos del proveedor (150 caracteres)

d) Góndola de exposición (3 dígitos)

Este archivo está ordenado por código de artículo creciente.

Además se dispone de otro archivo VENTAS con la información de ventas donde cada registro contiene:

a) Número de ticket (8 dígitos)      b) Código de artículo      c) Unidades (4 dígitos)

Este archivo sólo está ordenado por número de ticket y puede haber uno o varios registros por cada artículo de un mismo ticket.-

Desarrollar estrategia, algoritmo y codificación del programa que:

1) Imprima (ordenado por góndola y artículo), los artículos y cantidad a reponer (igual a suma de ventas), según el siguiente formato:

REPOSICION EN GONDOLA NUMERO 999					
ARTICULO	CANTIDAD	ARTICULO	CANTIDAD	ARTICULO	CANTIDAD
XXXXXXX	9999	XXXXXXX	9999	XXXXXXX	9999

2) Regenera el archivo VENTAS, dejándolo ordenado por ticket y artículo (un único registro con total de unidades), colocando después del último artículo del último ticket, registros con número de ticket = 0 y artículo = 'bbbbbbb' hasta el fin del archivo.

Nota: No hay espacio en disco. Tamaño máximo de nodo = 12 bytes. Memoria estática = 4000 bytes.

19/02/99

La D.G.I. requiere realizar un muestreo con las ventas de un año de un máximo de 50.000 empresas, y para ello cuenta con:

Un archivo de empresas EMPRE, ordenado por CUIT, donde cada registro contiene:

a) CUIT de la empresa (9 dígitos)      b) Razón Social (30 caracteres)      c) Domicilio (35 caracteres)  
d) Código postal (4 dígitos)      e) Localidad (25 caracteres)

Un archivo TOTMENS con las ventas mensuales (si hubo) de las empresas (ordenado por mes y CUIT), donde cada registro contiene:

a) CUIT de la Empresa      b) Número de mes (1 a 12)      c) Importe total de ventas del mes (real)

Solo hay ventas de empresas que están registradas en el archivo EMPRE.

Desarrollar la metodología necesaria para obtener un programa que:

1) Imprima un listado de las empresas que superen el promedio de ventas anual (con orden decreciente de importe) con título:

CUIT	DOMICILIO	LOCALIDAD	COD.POS TAL	IMP. ANUAL
999999 999	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XX	XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX XX	9999	99999999 99

2) Para las empresas con venta anual superior a \$1.000.000 genere un archivo (orden código postal y CUIT) con el siguiente registro:

IMPORTE TOTAL DE VENTAS DEL MES													
a) CUIT	b) Cod. Postal	c) 1	d) 2	e) 3	f) 4	g) 5	h) 6	i) 7	j) 8	k) 9	l) 10	m) 11	n) 12

Nota: Cantidad de bytes en disco 4.000.000 bytes. Memoria estática 82.000 bytes.

Memoria din mica 5.000.000 bytes. Promedio = suma de todos los b.3) / cant. total de empresas

26/02/99

En un megamercado de ventas mayoristas requieren estadísticas de ventas de un mes para los artículos promocionados, a partir de los archivos:

De artículos ARTIC.DAT (orden artículo) donde cada registro contiene:

a) Número de artículo (6 dígitos)      b) Descripción (25 caracteres)      c) Número de rubro (3 dígitos)  
d) Precio por unidad (real)      e) Código de promoción (S/N)      f) Stock en unidades (6 dígitos)

De clientes CLIE.DAT (orden C.U.I.T.) donde cada registro contiene:

a) CUIT (9 dígitos)      b) Razón Social (30 caracteres)  
c) Domicilio (50 caracteres)      d) Código postal (4 dígitos)

De ventas diarias VENT.DAT (orden número de factura) con un registro por cada ítem de la factura, cada uno de los cuales contiene:

a) Número de factura (5 dígitos)      b) Fecha (AAMMDD)      c) CUIT  
d) Número de artículo      e) Cantidad de unidades (4 dígitos)

Desarrollar la estrategia, algoritmo y codificación del programa que actualice el stock e imprima:

- 1) Un listado de facturas ordenadas por CUIT y número de factura:

CUIT	RAZON SOCIAL	NUMERO DE FACTURA	IMPORTE
999999999	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	99999	999999.99
		99999	999999.99
999999999	XXXXXXXXXXXXXXXXXXXXXXXXXXXX	99999	999999.99

- 2) Un listado de los artículos promocionados de cada rubro, en orden decreciente de su total de ventas del mes.

PROMOCION MENSUAL DEL RUBRO 999	IMPORTE TOTAL	NUMERO DE ARTICULO
	999999.99	999999

Nota: Promoción máxima de 50 artículos por rubro. Máximo 50.000 clientes. Memoria estática 500.500 bytes. Memoria dinámica 2.000.000 bytes. No hay espacio en disco.

5/03/99

El Banco de la Provincia vendió terrenos en cuotas y requiere controles sobre la cobranza de las cuotas.

Para ello dispone de los siguientes archivos:

MAESTRO de cobranza (orden número de propietario y cuota) con los cobros realizados antes de este proceso, donde cada registro contiene:

- a) Número de propietario (7 dígitos)      b) Número de cuota (1 a 60)  
c) Importe pagado (9999.99)      d) Fecha de pago (AAMMDD)

COBRANZA del mes (ordenado por fecha) donde cada registro contiene:

- a) Número de propietario      b) Número de cuota      c) Importe pagado  
d) Fecha de pago      e) Número de Banco (8 dígitos)

Desarrollar la estrategia, algoritmo y codificación necesaria para obtener el programa que:

- Ingrese el número de cuota para control.
- Actualice el archivo MAESTRO con los cobros del mes en proceso.
- Imprima un listado de propietarios morosos con por lo menos una cuota paga con el siguiente formato:

NUMERO PROPIETARIO	CUOTAS PAGADAS SOBRE 99
	1-----10-----20-----30-----40-----50-----60
9999999	PPP PPPPPP PPPPPPPPPPPP
9999999	PPPPPPPPPPPPPPPPPP PPPP

- 4) Imprima la recaudación de cada banco en orden decreciente de la misma.

NUMERO DE BANCO	IMPORTE RECAUDADO
99999999	999999.99
99999999	999999.99

Nota: Tamaño máximo de nodo = 14 bytes. Memoria estática 10.000 bytes. Disco suficiente sólo para actualización. Bancos <=1.000.

26/05/99

Se debe procesar información para la elección de Presidente, para ello se cuenta con los siguientes archivos:

Archivo de provincias PROV.DAT (sin orden) donde cada registro contiene:

- a) Código de provincia (1 a 24)      b) Nombre de la provincia (20 caracteres)

Archivo de listas de los partidos políticos LISPART.DAT ordenado por número de lista, en el que cada registro de los 999 contiene:

- a) Número de lista (1 a 999)      b) Nombre de la lista (20 caracteres)  
c) Nombre del candidato (30 caracteres)      d) Total de votos (0 a 99999999)

Archivo de votantes por mesa MESAS.DAT (desordenado) en el que cada registro contiene:

- a) Código de provincia      b) Número de mesa (1 a 99999)      c) Número de lista  
d) Votos femeninos (0 a 999)      e) Votos masculinos (0 a 999)      f) Votos impugnados (0 a 999)

Desarrollar la estrategia, algoritmo y codificación del programa que:

- 1) Imprima un listado de votos ordenado por provincia y número de lista (solo las que tengan total mayor que cero) según se indica:

PROVINCIA: XXXXXXXXXXXXXXXXXXXX		TOTAL DE VOTANTES: 99999999			
-----LISTA-----		-----CANTIDAD DE VOTOS-----			
NUMERO	NOMBRE	FEMENINOS	MASCULINOS	IMPUGNADO	TOTAL
				S	
999	XXXXXXXXXXXXXXXXXXXX	9999999	9999999	9999999	9999999

- 2) Actualice en el archivo LISPART.DAT el campo total de votos (máximo una vez por cada registro)
- 3) Informe el nombre del Presidente si alguno de los candidatos supera el 50% de los votos válidos, en caso contrario imprima ELECCION SIN GANADOR)

Nota: No hay espacio en disco. Memoria estática máxima 4850 bytes. Memoria dinámica máxima 384.000 bytes. Sólo hay 10 candidatos a Presidente.