

Una consultor desea hacer una estadística sobre los cortes de luz en el país a lo largo del año. Para ello cuenta con un **archivo** (cortes.dat) con los siguientes datos.

hora inicial de corte (hhmm), hora final (hhmm), id de subestacion, día, mes, id de localidad sin orden.

Se cuenta con:

- un **vector de localidades** completo ordenado por id asociado(PUP) con los siguientes datos: *nombre de localidad, cantidad de habitantes.* → *ordenado por ID - localidad*

A los efectos de esta solución se debe considerar que todos los registros de corte se dan dentro del día registrado, es decir no hay cortes que comiencen un día y terminen en otro y existe un solo registro por localidad.

Se solicita emitir un listado con los siguientes datos:

MES

Cantidad de horas promedio de corte POR MES 9999 → $\frac{\text{AUM - HORAS}}{12}$

Localidad(nombre) total de horas de cortes unidades de la multa x

xxxxxxx

99999

999999

Se pide:

- 1) declarar todas las estructuras necesarias para resolver la impresión del listado (2 puntos)
- 2) imprimir el listado tal como se detalla arriba ordenado por MES Y LOCALIDAD con el promedio de horas de corte para cada mes y calcular el importe de multa de cada localidad considerando que menos de 3 horas no se calculan, entre 3 y 6 se calculan a razón de 8 unidades de multa la hora, y mas de 6 se calculan a razón de 12 unidades cada hora. Por ejemplo 3 horas de corte suman 24 unidades de multa y 8 horas sumarian 96.

nota: para imprimir el listado debe resolver el problema desarrollando funciones con los parametros que considere adecuados. Puede utilizar funciones de biblioteca, siempre que respete las firmas que se enseñaron en clase.

emitir listado con totales correctos 4 puntos, logica adecuada para el cálculo de multas 4 puntos.

^ Una empresa de servicios requiere procesar las transacciones realizadas por cada una de sus 20 sucursales.

El arquitecto de la solución requiere se almacenen los datos, provenientes de un archivo de registros en una estructura estatica de dos dimensiones, en las que las que una dimensión representan las sucursales (son 20) y la otra los días del mes (considerar 31). Para cada par (sucursal, dia) se deben establecer las transacciones (conteniendo detalle e importe) que se tienen registradas en el archivo siguiendo el orden de aparición del mismo.

Sucursal	Dia	detalle	importe	Otros datos
1..20	1..31	Cadena de 20 car	float	Cadena 30 car

SE requiere

- 1) Desarrollar al función *void generarEstructura (Archivo, estructura seleccionada)* que cargue en la estructura que decida los datos del archivo. Debe declarar y definie la estructura y justificar su seleccion
- 2) Desarrollar la Función *void mostrarImportesSucursal(estructura)* que informe el total de transacciones y su importe de cada una de las sucursales, ordenada por sucursal. Determine cual es la estructura que considera mas adecuada, justifique la elección y definala
- 3) Desarrollar la Función *float menorImporte(estructura)* que informe el menor de los importes de cada sucursal, (solo hay uno), ordenada por sucursal. Determine cual es la estructura que considera mas adecuada, justifique la elección y definala

2do Parcial RECUPERATORIO Algoritmos y estructura de datos 2023	
Apellido y Nombre:	Legajo:

Se quiere almacenar en memoria los gastos efectuados por cada uno de los 15 sectores de una empresa en un mes determinado para realizar un análisis estadístico.

Para ello se implementará una matriz de listas, donde las filas representan los sectores y las columnas los días y cada elemento es la lista con cada gasto de ese sector en ese día.

Los datos para armar la estructura se encuentran en el archivo de datos binario "Gastos.dat", donde cada registro representa un gasto efectuado y tiene:

- Número de sector (de 1 a 15)
- Día del mes (de 1 a 31)
- Detalle
- Importe gastado.

El archivo no está ordenado y puede tener varios gastos realizados por un mismo sector.

1) Función **cargarEstructura** debe generar la estructura en memoria con los datos del archivo, tener en cuenta que la estructura hay que utilizarla en las otras funciones. (2)

2) Función **mostrarGastos** debe mostrar por pantalla el siguiente listado: (5)

Sector 1

Día 1

..... (detalle e importe de cada gasto)

.....

Día 2

.....

Sector 2

.....

Detalle e importe de cada gasto (ordenado de mayor a menor por importe) agrupado por sector y por día.

3) Función **mayorGasto** debe mostrar qué sector y qué día se realizó el mayor gasto (suponer único) (3)