



Duración total del examen: 3 horas

Ejercicio 1 [4 puntos]

El ministerio de Medio Ambiente ha encargado a la Confederación Hidrográfica del Ebro (CHE) la gestión de los incendios que ocurren en las comunidades autónomas por las que pasa el río Ebro. Para llevar una mejor gestión de los incendios la CHE decide almacenar toda incidencia en un fichero.

Para ello, solicita a las comunidades autónomas que forman parte de la CHE que a lo largo del año registren los incendios en un fichero de registros con la siguiente estructura y ordenados por fecha en sentido ascendente (de mas antiguo a más reciente):

```
tpfecha=array [1..6] of char;  
{Las fechas se representan con 6 caracteres, dos caracteres para el año, 2  
para el mes y dos para el día, en este orden. Por ejemplo la fecha de hoy  
se representa 040624}  
  
tpIncidencia=record  
    localidad: array[1..24] of char;  
    fecha:tpfecha;  
    hectareas:integer;  
    victimas:integer;  
    ccaa:tpccaa;  
end;
```

Los ficheros de registros de las seis comunidades se van a mezclar, resultando un único fichero de registros ordenados por fecha y en sentido ascendente.

Se pide:

Programa pascal que realice la mezcla de los 6 ficheros de registros resultando el fichero de registros ordenado por fecha.

Los ficheros de datos a mezclar tienen los siguientes nombres: **santander**, **la_rioja**, **navarra**, **pais_vasco**, **aragon** y **catalunya**. El fichero resultante de la mezcla se llamará **incendiosEbro**. Con objeto de facilitar el tratamiento de los ficheros se declara el tipo **tpcaa**, y la función **ccaaTostring** que dado un valor de comunidad autónoma devuelve la cadena con el nombre del fichero que contiene los registros de la comunidad correspondiente.

```
tpccaa=(santander, la_rioja, navarra, pais_vasco, aragon, catalunya);  
Function ccaaTostring(ccaa:tpccaa):tpnombre20;  
{Esta función será útil para asociar a ficheros externos. Iterando con  
todos los valores de tpcaa se obtienen los nombres de los ficheros.}  
begin  
    case ccaa of  
        santander: ccaaTostring:='santander';  
        la_rioja: ccaaTostring:='la_rioja';  
        navarra: ccaaTostring:='navarra';  
        pais_vasco: ccaaTostring:='pais_vasco';  
        aragon: ccaaTostring:='aragon';  
        catalugna: ccaaTostring:='catalunya';  
    end;
```

NOTA:

- Para “recorrer” todos los ficheros puedes utilizar los tipos

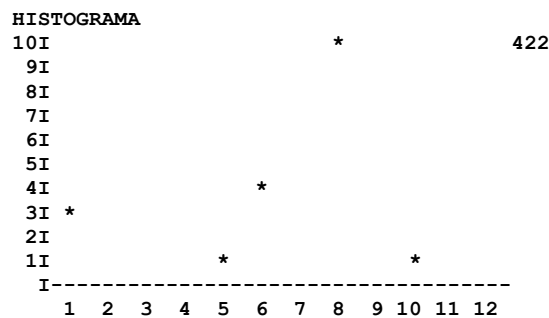
```
tpfichero_Incidencias=file of tpIncidencia;
```

```
tpficheros_ccaa=array[tpccaa] of tpfichero_Incidencias;
```

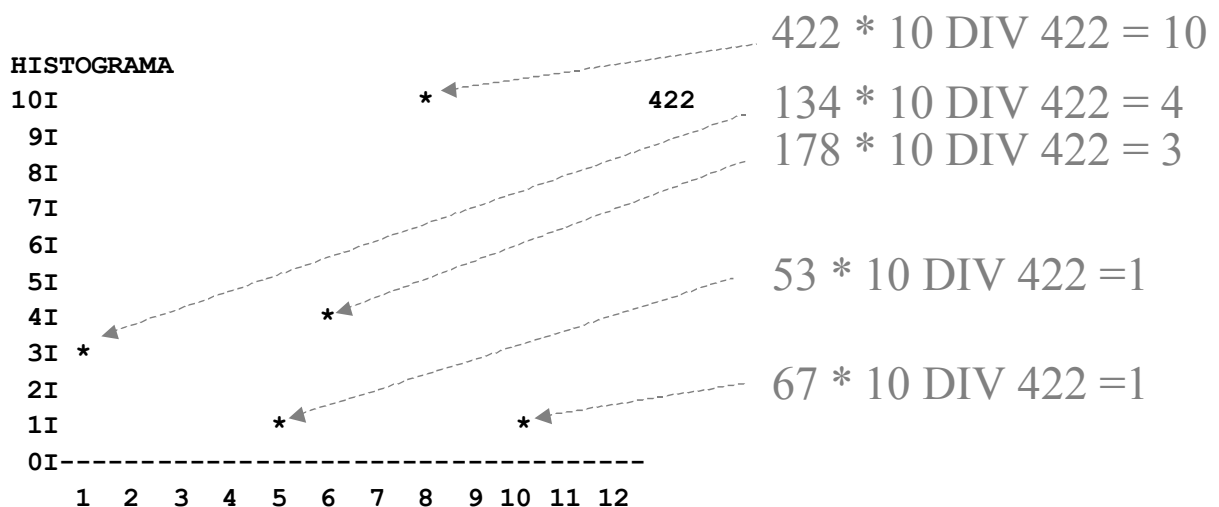
Ejercicio 2 [3 puntos]

Se pide escribir un **procedimiento Pascal** que dibuje por pantalla **un histograma** con el número de hectáreas quemadas en toda la cuenca por cada mes a partir del fichero de registros **incendiosEbro** creado por el programa del ejercicio 1. Las alturas del histograma se ajustan de modo que el valor máximo se representa con 10 unidades (en vertical) y el resto se ajusta de modo proporcional (véase ejemplo).

El histograma indicará el valor máximo de forma numérica en la primera línea. La gráfica debe ser como la del ejemplo del resultado de ejecución del programa que se muestra a continuación:



- El anterior histograma resulta de los siguientes valores ejemplo:



Ejemplo de histograma para los valores siguientes de hectáreas quemadas

Ene :134 Feb: 0 Marz: 0 Abr: 0 May: 53 Jun: 178 Jul: 0 Ago:422 Sep: 0 Oct: 67 Nov: 1 Dic: 36



**Centro Politécnico Superior
Universidad de Zaragoza**

Duración total del examen: 3 horas

Ejercicio 3 [3 puntos]

Tras el sorteo de los grupos que conforman la primera fase de la próxima Eurocopa de fútbol, algunas federaciones hicieron públicas sus dudas sobre el resultado del mismo. La sombra de un posible tongo ha llevado al máximo organismo futbolístico europeo, la UEFA, a desarrollar un nuevo sistema informático para la realización de futuros sorteos.

Como parte del nuevo sistema informático se han declarado estructuras de datos para la realización del sorteo y el almacenamiento de los resultados. Primeramente, se describen las estructuras empleadas en la realización del sorteo:

const

```
maxCabezasSerie = 4;  
maxSegundasCabezaSerie = 4;  
maxRestoEquipos = 8;
```

type

```
tpEquipo = string[50];  
  
tpCabezasSerie = 1..maxCabezasSerie;  
tpSegundasCabezaSerie = 1..maxSegundasCabezaSerie;  
tpRestoEquipos = 1..maxRestoEquipos;  
  
tpBombos = record  
  bomboCabezasSerie: array[tpCabezasSerie] of tpEquipo;  
  bomboSegundosCabezaSerie: array[tpSegundasCabezaSerie] of tpEquipo;  
  bomboRestoEquipos: array[tpRestoEquipos] of tpEquipo;  
end;
```

En la estructura **tpBombos** están contenidos los nombres de las 16 selecciones participantes en el torneo organizadas según su categoría (esta categoría se les atribuye en función de su clasificación final en competiciones previas). Existen tres categorías diferentes: cabezas de serie, segundas cabeza de serie o resto de equipos. En el campo **bomboCabezasSerie** de la estructura **tpBombos** se almacenan los nombres de las 4 selecciones que son cabeza de serie de la Eurocopa; en el campo **bomboSegundosCabezaSerie**, los nombres de las 4 que son segundas cabeza de serie; y en el campo **bomboRestoEquipos**, los nombres de las 8 restantes selecciones. En total, entre los 3 bombos, están las 16 selecciones participantes que van a ser sorteadas posteriormente. No es relevante el orden en el que están almacenadas las selecciones dentro de cada bombo.

Por otro lado, las estructuras para almacenar los resultados del sorteo son las siguientes:

const

```
maxEquipos = 4;
```

type

```
tpEquiposGrupo = 1..maxEquipos;  
tpGrupo = array [tpEquiposGrupo] of tpEquipo;  
tpLetraGrupo = (grupoA, grupoB, grupoC, grupoD);  
tpEurocopa = array [tpLetraGrupo] of tpGrupo;
```

Se pide implementar el siguiente procedimiento:

Procedure sorteoGrupos(var bombo: tpBombos; var resultado: tpEurocopa)

{Este procedimiento sortea la composición de cada grupo de la competición (4 en total) a partir de las selecciones participantes del parámetro 'bombo'. Cada grupo deberá estar compuesto por una selección 'cabeza de serie' (se almacenará en la componente 1 del correspondiente vector del tipo tpGrupo), una 'segunda cabeza de serie' (componente 2) y dos 'resto de equipos' (componentes 3 y 4). La distribución de selecciones-grupo será totalmente aleatoria (evidentemente, cada selección formará parte de un único grupo).}

NOTAS:

- **Se supone predefinida la función**

```
Function Random (num:integer):integer;  
{Pre: Num >0}  
{Post: Devuelve aleatoriamente un número entre 0 y num-1}
```

- **Puedes añadir las estructuras de datos adicionales que estimes oportuno o modificar las propuestas para resolver mejor el problema planteado.**