

Apellidos _____

Nombre _____

Grupo _____

--	--	--	--	--	--	--	--

- | | |
|---|--|
| <ul style="list-style-type: none"> • Duración del examen: 3h 30 min. • Poner el nombre y apellidos en todas las hojas del examen. • Estas hojas deben entregarse SIEMPRE. • Conteste, en esta misma hoja, en el espacio que se ha dejado, a los ejercicios 5 y 6. | <ul style="list-style-type: none"> • Realizar cada problema en páginas diferentes • Se valorará la presentación y la claridad en la solución • Se valorará la adecuación de las estructuras utilizadas al problema a resolver. • No se calificarán las respuestas escritas en lapicero |
|---|--|

1. (4,5 pts.) En el archivo *FotosVendidas.dat* se encuentra almacenada información referida a las ventas de fotos de un estudio fotográfico. La información que se mantiene por cada foto vendida es: un identificador de foto, único para cada foto; el precio de venta de la foto; un identificador del autor de la foto, único para cada autor; el nombre del autor de la foto; y el nombre del comprador de la misma.

Las estructuras de datos que se han definido son:

```

Type Tfoto =          record          IdFoto : integer;
                                Precio : real;
                                IdAutor: 1..30;
                                Autor  : string;
                                Comprador : string;
                                end;
TFicheroFotos = file of Tfoto;

```

Se sabe, además, que el archivo de las fotos, *FotosVendidas.dat*, está ordenado por el campo *Comprador* y que hay información relativa a 30 autores distintos.

Se pide:

- (2 pts.) Elaborar un programa en *Pascal* que escriba en un fichero de texto, *estadísticas.txt*, a razón de una línea por *comprador*, el nombre del comprador, el número total de fotos que ha comprado y el precio total que ha pagado.
- (2 pts.) Elaborar un programa en *Pascal* que presente por pantalla el nombre de los tres *autores* que más fotos han vendido junto con el número total de fotos que ha vendido cada uno de ellos (si hay empate entre varios, sólo se imprimirán tres de ellos).
- (0'5 pts.) ¿Cómo podría elaborarse un programa que realizase ambas tareas mediante un único recorrido del fichero?

2. (1 pt.) Dado el siguiente programa en *Pascal*, se pide:

- ¿Cuál es el resultado del siguiente programa?
- Escribir una versión no recursiva de *h*.

```

program misterioso (output);
function h(n:integer): integer;
begin
    if odd(n) then h := -h(2*n)
                else h := 2*n;
end;

function g(n:integer): integer;
begin
    g := n+n+h(n);
end;

var i:integer;
begin
    for i:=1 to 5 do write (g(i), ' ');
end.

```

3. (1'25 pts.) Elaborar un subprograma que obtenga el conjunto de vocales que aparecen en una cadena de caracteres dada (*string*). En la cadena no hay tildes ni diéresis, y se considera que 'a' y 'A' son la misma vocal.
4. (1'25 pts.) Dada una matriz de caracteres con índices [1..N, 1..N], elaborar un subprograma que obtenga, como cadena de caracteres, la concatenación de los que ocupan la diagonal secundaria de dicha matriz.
5. (1 pt.) Supóngase que se tiene definido:

```
Const I1=... ; IP=... ; UNVALOR=... ; (* ciertos valores *)
Type
  TElemento = ... ; (* un tipo *)
  TVector = array [I1 .. IP] of TElemento; (* I1 <= IP *)
```

y que se tiene a continuación el código que aparece al final de este ejercicio.

- a. Para las variables que existen cuando el programa en ejecución se encuentra en el punto marcado (**), descríbanse sus nombres, valores y tipos respectivos en el espacio siguiente:

Nombre o forma de acceso	Tipo	Valor

- b. Escríbase, en el espacio que sigue a (**), el código necesario para iniciar todas las componentes del vector a UNVALOR.

```
var p1, p2 : ^TElemento;
    p3 : ^TVector;
    n : I1 .. IP;
BEGIN
  new (p1); new (p2); new (p3);
  p1^ := UNVALOR;
  p2 := p1;
(**)
```

END.

6. (1 pt.) Indicar con V o F la veracidad o falsedad de las siguientes afirmaciones, en el lenguaje *Pascal*.

- [] Los tipos *array* son estructuras de datos homogéneas.
- [] El tipo base de un *array* debe ser obligatoriamente simple.
- [] Los tipos índice de un *array* pueden ser cualesquiera simples.
- [] El tamaño de un *array* puede cambiarse en tiempo de ejecución.
- [] Se puede escribir todo un *array* con una única sentencia de escritura (*write*) en la pantalla.
- [] Se puede escribir todo un *array* con una única sentencia de escritura (*write*) en determinados tipos de archivo (*file*).
- [] Los *punteros* son variables dinámicas.
- [] Un *parámetro formal* no puede tener el mismo identificador que una *variable global*.
- [] Los *arrays* deben ser pasados como parámetro obligatoriamente por variable.
- [] Los *ficheros (file)* deben ser pasados como parámetro obligatoriamente por variable.