

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. | <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. (2,5 pts.) En el archivo *Asignaturas.dat* se encuentra almacenada información referida a las asignaturas ofertadas en la Universidad de Valladolid. La información que se mantiene para cada asignatura es: código de la asignatura, descripción, número de créditos totales, número de créditos teóricos, número de créditos prácticos y código de la titulación en la que se imparte. La estructura del registro y del archivo en los que se mantiene la información es:

```
Type TAsignatura = record
    CodAsignatura : integer;
    Descripción : string;
    NCreditosTotales : real;
    NCreditosTeoricos : real;
    NCreditosPracticos : real;
    CodTitulacion : integer;
end;
TFicheroAsignaturas = file of TAsignatura;
```

Los registros están almacenados en el archivo sin ningún orden preestablecido; se sabe que el número máximo de asignaturas que oferta la Universidad es de 1.500 y que no hay ninguna repetida (es decir, el código de la asignatura es único y no se repite).

Se desea crear un *índice auxiliar* que permita saber en qué posición del archivo *Asignaturas.dat* se encuentra la información de cada una de ellas. En este *índice* se debe almacenar, para cada asignatura, la posición que ocupa en el archivo, de forma que la información esté ordenada por el código de la asignatura. La estructura a utilizar será:

```
Type TIndice = record
    CodAsignatura : integer;
    Posicion : integer;
end;
```

Finalmente, el *índice* debe guardarse en un archivo del tipo `TFicheroIndice = file of TIndice`.

Por, ejemplo:

12235	Programación I	7,5	4,5	3	223
10268	Física I	7,5	4,5	3	224
16845	Bases de Datos	7,5	4,5	3	223
12136	Economía	4,5	3	1,5	223
11598	Programación II	7,5	3	4,5	223

Fichero de las Asignaturas

10268	2
11598	5
12136	4
12235	1
16845	3

Fichero del Índice

Se pide elaborar un programa en *Pascal* que se encargue de construir el índice auxiliar y de escribirlo en un archivo adecuado, a partir de la información existente en el archivo *Asignaturas.dat*, del que no hace falta comprobar su existencia.

2. (2,5 pts.) Dado el tipo de datos `TTexto = Array [1..n] of char`; donde n es una constante menor que 255, y un texto en castellano, formado por palabras con las siguientes características:

- los espacios y las comas separan palabras,
- el texto es de longitud variable,
- el tamaño máximo del texto es n ,
- el texto termina siempre en un punto,
- el texto está almacenado en una variable del tipo `TTexto`,

Elaborar un subprograma que analice dicho texto y obtenga la palabra que, teniendo el mayor número de vocales distintas, sea la más larga (si hay varias palabras que cumplan esta condición, se obtendrá sólo una de ellas).

Por ejemplo, dado el texto “La gravísima crisis social que vive Bolivia se ha cobrado la dimisión del presidente, Carlos Mesa.”, la palabra pedida es “Bolivia”.

3. (2 Pts.) Un saltamontes loco vive saltando sobre un plano de coordenadas x (oeste-este) e y (sur-norte), de forma que aterriza siempre sobre puntos de coordenadas enteras. Su movimiento sigue la pauta que se detalla a continuación: Si está en una posición (x, y) , y x es par, calcula como destino el punto $(x', y') = (x/2, y-x)$; de lo contrario, calcula $(x', y') = (y, 3x+1)$. Si la distancia a la que tiene que saltar es mayor que 25 unidades ($distancia = ((x' - x)^2 + (y' - y)^2)^{1/2}$), divide ambas coordenadas del destino que ha calculado por la mitad (división entera) las veces que sea necesario para que la distancia sea menor o igual a 25. Entonces salta a esa posición. Realiza un salto por segundo.

Elabore una función recursiva en Pascal tal que, dadas dos posiciones del plano (x_1, y_1, x_2, y_2) y un número de segundos s , determine si es posible que, el saltamontes, partiendo de la posición (x_1, y_1) llegue a la posición (x_2, y_2) en s segundos o menos.

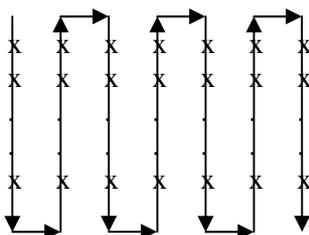
4. (1,5 Pts.) Realice la traza del siguiente fragmento de programa, usando los símbolos “-“ para indicar una variable que no existe y “?” para indicar un valor desconocido. Además deben indicarse los errores y las acciones no recomendables que hay en el fragmento. Aunque haya errores, considérese que el programa continua su ejecución.

```

VAR
    i, x, y: integer;
    p, q: ^integer;
BEGIN
    x := 8;
    y := 2;
    p := q;
    new (p);
    ^p := y;
    q := p;
    for i:=1 to ^p do
        x := x+1;
    dispose (p);
    x := ^q;
    new (q);
    new (p);
    q := p;
    ^p := x;
    ^q := y;

```

5. (1,5 Pts.) Dado el tipo de datos `TMatriz = Array [1..n, 1..m] of char`, donde m y n son constantes cuyo producto ordinal ($m \times n$) no supera el valor 255, escribir un subprograma que descargue el contenido de una matriz de este tipo en una cadena de caracteres en el orden señalado por el diagrama.



Por ejemplo, la matriz

$$\begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}$$

debería transformarse en la cadena: ‘adebcf’