Práctica Evaluable Tema 4.

Arrays, estructuras y cadenas de texto

Objetivos.

- Repasar los conceptos estudiados hasta ahora.

Consideraciones iniciales.

- En cada ejercicio se indica su puntuación, incluyendo la puntuación de cada subapartado.
- Cada ejercicio tiene que estar en un fichero con el nombre indicado en el enunciado y debe contener una clase con ese mismo nombre, pero sin la extensión ".cs"

Código implementado

Para cada archivo fuente entregado se deberá incluir como comentario en las primeras líneas del archivo el nombre del autor, la indicación de la práctica evaluable y el número del ejercicio.

Además se incluirá un listado de todos los apartados, indicando si han sido implementados totalmente, parcialmente o no ha sido realizado.

Por ejemplo:

```
/*
Perez Gomez, Andres
Practica Evaluable Tema 4
Ejercicio 1
Apartado 1.1  si / no / parcialmente
Apartado 1.2  si / no / parcialmente
Apartado 1.3  si / no / parcialmente
Apartado 1.4  si / no / parcialmente
...
*/
```

```
/*
Perez Gomez, Andres
Practica Evaluable Tema 4
Ejercicio 2
Apartado 2.1  si / no / parcialmente
Apartado 2.2  si / no / parcialmente
Apartado 2.3  si / no / parcialmente
...
*/
```

Entrega.

Se debe entregar un archivo comprimido ZIP con los archivos fuente (extensión .cs) de los ejercicios propuestos.

- Nombre del archivo: **Apellidos_Nombre_PracT4.zip**

  Por ejemplo, si te llamas Andrés Pérez Gómez el archivo debe llamarse *Perez_Gomez_Andres_PracT4.zip*

Desarrollo.

Ejercicio 1.

Nombre del fichero: "Alumnos.cs"
Puntuación máxima: 10 puntos

Diseña un programa para gestionar un array de alumnos, con capacidad para hasta 90 alumnos.

- **1.1 (1 punto)** Cada registro está compuesto por una estructura para almacenar los siguientes datos. Elige el tipo de dato que mejor se ajuste a cada caso:
  - nombre
  - apellidos
  - ciudad
  - fecha nacimiento: otra estructura para almacenar día, mes y año de nacimiento
  - notas: otra estructura para almacenar las notas de 6 prácticas, 3 exámenes (1ª, 2ª y 3ª evaluación) y nota final.

El programa presentará continuamente al usuario un menú para elegir estas opciones:

- Añadir un alumno.

  1.2 (0,5 puntos) Solicitar por consola los datos correspondientes al alumno e introducirlo en la primera posición libre al final de las existentes.

  Para la nota final no se solicitará ningún dato al usuario, se calculará internamente en el programa de la siguiente forma:

  - Nota Final = 20% eva1 + 30% eva2 + 50% eva3
  - Nota Eva1 = 30% ((practica 1 + practica 2) * 0,5) + 70% examen
  - Nota Eva2 = 30% ((practica 3 + practica 4) * 0,5) + 70% examen
  - Nota Eva3 = 30% ((practica 5 + practica 6) * 0,5) + 70% examen

  No se ordenará el array después de esta operación.

  A la hora de introducir el nombre y apellidos, y la fecha de nacimiento, se os proponen dos opciones para implementarlo: una elaborada (A) y otra sencilla (B).

  Opción A:

  1.2.1 (0,75 puntos) La petición por consola del nombre y los apellidos se almacenará en una única variable de tipo “string”. Dentro del programa se separá el valor de esta variable en otras 2 variables, una para el nombre y otra para los apellidos. Por simplicidad, se asume que cada alumno tendrá un nombre simple y no compuesto.

  Por ejemplo, “José García Pérez” y no “José Antonio García Pérez”. El resultado del tratamiento de “José García Pérez” será: nombre=”José” y apellidos=”García Pérez”.

  1.2.2 (0,75 puntos) La petición por consola de la fecha de nacimiento se hará en un única variable de tipo “string” con los campos separados por “-”. Dentro del programa se separá el valor de esta variable en otras 3 variables, una para el día, otra para el mes y otra para el año.

  Por ejemplo, el string fecha con valor “15-11-2009” el programa lo convertirá en dia=15, mes=11 y año=2009.

Opción B:

1.2.3 (0 puntos) La petición de datos de nombre, apellidos, día, mes y año se podrá realizar de forma individualizada en variables independientes, en cuyo caso, no se puntuará la parte correspondiente.

- **1.3 (1 punto)** Borrar un alumno.

  Primero se mostrará un listado de los alumnos del array y se pedirá al usuario que elija la posición a borrar. Los elementos que están después de esa posición, deberán desplazarse "hacia la izquierda" para que no queden huecos, tal y como se explica en los apuntes. No se ordenará el array después de esta operación.

- **1.4 (0,7 puntos)** Mostrar los alumnos de una ciudad.

  Se mostrarán todos los datos de los alumnos cuya ciudad coincida con la indicada, ignorando mayúsculas y minúsculas, y teniendo en cuenta coincidencias parciales.

  Por ejemplo, si el usuario busca por "Ma", le deberán salir todos los alumnos de Madrid, o de Marbella, o de Almansa.

- **1.5 (0,3 puntos)** Mostrar los alumnos nacidos en un año concreto.

  Mostrar todos los datos de los alumnos cuyo año de nacimiento coincida con el indicado.

- **1.6 (1 punto)** Ordenar alfabéticamente el array.

  Se ordenará el array por los apellidos, de menor a mayor, usando el método de ordenación que se desee de los expuestos en los apuntes. Se mostrará por pantalla el resultado ordenado.

- **1.7 (0,75 punto)** Ordenar el array por mayor nota final.

  Se ordenará el array en orden descendente por la nota final obtenida, primero el de mayor nota. Se mostrará por pantalla el resultado ordenado.

- **1.8 (1,5 puntos)** Calcular porcentajes aprobados y suspensos.

  El programa calculará y mostrará el porcentaje de aprobados y suspensos de cada uno de las notas (prácticas, exámenes y nota final. Junto a los porcentajes también se deberán los datos numéricos y el número total de alumnos.

  No es necesario que el array esté ordenado, se pueden calcular los porcentajes con el array desordenado.

- Salir del programa

1.9 (1 punto) Control de errores

Se deben controlar los posibles errores en la introducción de datos, y volverlos a pedir al usuario en caso de datos incorrectos. Algunos errores típicos de introducción de datos son: teléfonos con caracteres diferentes de números, fechas con día, mes o año no válidos...

Por simplicidad para la realización del programa:

- No se comprobará que la fecha de nacimiento sea válida completamente. Se asumirá que todos los meses tienen 31 días. Se deberá comprobar que el mes de nacimiento está comprendido entre el 1 y el 12. Tampoco se puede haber nacido antes del año 1900 ni con posterioridad al 2010.

Observaciones

- **1.10 (0,5 puntos)** Además de la implementación del programa siguiendo los pasos indicados, se deberá tener en cuenta la USABILIDAD del mismo, es decir, que el usuario que lo emplee sepa en todo momento lo que tiene que introducir, y se le muestre la información adecuada. Por ejemplo, antes de pedirle un dato al usuario, debemos indicarle qué dato debe introducir.
- Al mostrar por pantalla el array completo se debe poner especial cuidado que los datos estén bien formateados en una sola línea. Si fuera el caso, dependiendo del listado a mostrar, algunos campos no sería necesario mostrarlos. Por ejemplo, si se muestra el array ordenador por nota final, no sería necesario mostrar el resto de notas.
- **1.11 (0,25 puntos)** En cada listado que se muestre al usuario, las posiciones en el array deben numerarse desde la 1 (el usuario puede no entender qué significa la posición 0). Sin embargo, internamente, los datos se deben almacenar desde la posición 0. Se deberá entonces establecer una "traducción" entre lo que se muestra al usuario (o la posición que el usuario elija, en el caso del borrado) y la posición real del dato en el array.
- Observad que si optáis por la opción B para implementar “Añadir alumnos”, la suma total de puntos para el ejercicio es de 8,5 puntos como máximo, mientras que con la opción A se alcanzan los 10 puntos (máximo) del ejercicio.