

Programación 2

Examen de teoría (2020 - C3)

3 de junio de 2020



Ejercicio 1 (6 puntos)

Instrucciones

- Duración: **1 hora y 30 minutos**. La hora tope para la entrega de este primer ejercicio será las **10:30**
- Después de esto, habrá un **descanso de 15 minutos** y a continuación se podrá descargar el enunciado del siguiente ejercicio
- En este primer ejercicio debes entregar dos ficheros: un fichero `ej1.cc` que contendrá el código fuente del ejercicio y otro `entrada.txt` que incluirá un ejemplo de fichero de entrada de texto, tal y como se describe más abajo en el enunciado. Pon tu DNI y tu nombre en un comentario al principio del fichero `ej1.cc`
- La entrega se realizará como en las prácticas, a través del servidor del DLSI (<http://pracdlisi.dlsi.ua.es>), en el enlace **Programación 2 - Ingeniería Informática**. Puedes realizar varias entregas, aunque sólo se corregirá la última
- Está permitido durante el examen consultar los materiales de la asignatura, libros, apuntes y/o referencias en Internet. Si el código **no compila**, se restará un punto de la nota final del ejercicio
- Para la resolución de los ejercicios, **solo está permitido el uso de las librerías y funciones vistas en clase de teoría y prácticas**. No es válido aportar soluciones copiadas y pegadas de Internet

Código de conducta

- El examen es un trabajo individual. **Cualquier indicio de copia, comunicación con otros alumnos (p.ej. mediante grupos de WhatsApp) o intervención de terceras personas en su realización será sancionada según la legislación vigente**, con medidas que pueden llevar a la **expulsión** del alumno/a de la titulación

Enunciado

Debes desarrollar un programa en C++ que gestione información de los clientes de una empresa de telecomunicaciones. Habrá dos campos de información que obligatoriamente tendrás que almacenar para cada cliente:

- DNI: un valor entero sin signo (`unsigned`)
- Nombre y apellidos: una cadena de caracteres (`string`)

Además de esos dos campos, tendrás que almacenar otros tres campos más de información sobre cada cliente, los que tú decidas (por ejemplo, la fecha de alta).

Crea un registro (`struct`) con esos cinco campos para manipular en tu programa los datos de los clientes. Entre los cinco campos tienes que usar al menos tres tipos de datos diferentes (por ejemplo, `unsigned`, `string` y `float`).¹

Introduce los comentarios que creas necesarios en el código para que el profesor que te corrija entienda la información que estás almacenando y, en general, para aclarar cualquier función que desarrolles.

Tu programa deberá leer los datos de los clientes almacenados en un fichero de texto y almacenarlos todos en memoria, mediante un `vector` STL, para poder manipularlos luego. El formato del fichero de texto será el que tú decidas, mientras te permita almacenar los cinco campos que hayas definido en el registro de cliente. Junto con el código fuente de este ejercicio, deberás entregar un fichero `entrada.txt` que contenga un ejemplo de fichero de entrada para tu programa, con la información de al menos tres clientes.

¹A estos efectos, consideramos que `unsigned` e `int` son el mismo tipo de dato.

No es necesario comprobar errores en el formato del fichero (por ejemplo, que falte algún campo o haya líneas en blanco). Puedes asumir que siempre será correcto.

Una vez leído y almacenado en memoria el contenido del fichero de texto, el programa deberá realizar una serie de transformaciones en los datos y escribirlos en un fichero binario. Para ello, define un nuevo `struct` que te permita guardar la información de los clientes en el fichero binario siguiendo el formato que tú decidas.

Antes de guardar la información en el fichero binario, deberás realizar estas tres operaciones:

- Pasar a mayúscula la primera letra del nombre y apellidos del cliente. Por ejemplo, si el nombre leído es `arturo manuel del olmo soriano`, deberá transformarse en `Arturo Manuel Del Olmo Soriano`
- Si el DNI del cliente es uno de los siguientes, no se deberá de guardar su información en fichero (se descartará sin más): 47859201, 72345612 y 23487902
- Aplicar algún tipo de transformación a alguno de los campos adicionales que has definido en el fichero de texto. Por ejemplo, si has definido un campo `fecha` que tiene un formato `dd-mm-aaaa` (ej. "20-02-2019", podrías transformarla al formato `aaaammdd` antes de guardarla en el fichero binario (ej. "20190220"). Este ejemplo de transformación no lo puedes usar en tu solución. Tienes que pensar uno propio. Recuerda poner los comentarios que sean necesarios en el código para que el profesor que te corrija entienda qué tipo de transformación estás haciendo

Tanto a la hora de leer el fichero de texto, como a la hora de escribir el fichero binario, deberás comprobar que se han podido abrir correctamente. Si no es así, deberás mostrar un mensaje de error y finalizar el programa.

El nombre del fichero de texto y el del fichero binario se pasarán por línea de comando como argumentos de programa. El fichero de texto irá precedido de la opción `-i` y el binario de la opción `-o`. Ejemplos válidos de ejecución:

```
$ ej1 -i ficheroEntrada.txt -o ficheroSalida.bin
```

```
$ ej1 -o ficheroSalida.bin -i ficheroEntrada.txt
```

Si apareciera algún parámetro incorrecto, o faltara alguno de los parámetros o nombres de fichero, habría que mostrar un error de parámetros incorrectos y terminar el programa.