

Programación 2

Examen de teoría (2020 - C4)

14 de julio de 2020



Ejercicio 1 (5,5 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 un único fichero, `ej1.cc`, que contendrá el código fuente del ejercicio. Pon tu DNI y tu nombre en un comentario al principio del fichero
- Junto con este enunciado se te han proporcionado tres ficheros de ejemplo: `products.bin` (fichero binario de productos), `types.bin` (fichero binario de tipos de producto) y `output.txt` (fichero de texto generado a partir de los dos ficheros binarios anteriores)
- La entrega se realizará como en las prácticas, a través del servidor del DLSI (<http://pracdlsi.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

Uno de tus clientes va a inaugurar una tienda *online* para vender material de oficina. Debes desarrollar un programa en C++ que le permita exportar su inventario de productos a un fichero de texto para luego poder incorporar esa información a su página web. Los datos del inventario están guardados en dos ficheros binarios: `products.bin` y `types.bin`.

El fichero `products.bin` contiene todos los productos del cliente, cada uno de ellos almacenado en un registro con la siguiente información:

- El número de referencia: `char [16]` (Ej. BA45CC124GB128I)
- El nombre del producto: `char [50]` (Ej. Papel Navigator)
- La descripción del producto: `char [200]` (Ej. Papel multiusos para impresoras - 2500 hojas)
- El código de tipo de producto: `unsigned` (Ej. 42)
- El precio del producto: `float` (Ej. 23.85)
- Un valor booleano que indica si es un producto nuevo esta temporada o no: `bool` (Ej. `true`)

- El número de unidades de este producto que el cliente tiene en su almacén: `int` (Ej. 145)

El fichero `types.bin` contiene los tipos diferentes de productos que hay, cada uno de ellos almacenado en un registro con la siguiente información:

- El código de tipo de producto: `unsigned` (Ej. 42)
- La descripción del tipo: `char [200]` (Ej. `Papel para fotocopadoras y multifunción`)

En los ejemplos dados arriba, puedes ver que el producto que hemos puesto como ejemplo en el fichero `products.bin` tiene de código de tipo de producto 42. En el fichero `types.bin` nos dice que la descripción del tipo de producto con código 42 es `Papel para fotocopadoras y multifunción`.

El programa debe leer estos dos ficheros binarios y exportar los campos leídos a un fichero de texto. En ese fichero, la información de cada producto se almacenará en una línea distinta. Cada campo de información de un producto estará delimitado por un carácter separador: coma, punto y coma, o dos puntos. El tipo de separador a utilizar se indicará a través de un argumento por línea de comando, tal y como se describe más abajo.

Los campos que se guardarán para un producto en el fichero de texto serán los siguientes:

- El número de referencia
- El nombre del producto
- La descripción del producto. En este caso, como máximo deberás exportar 100 caracteres de los 200 posibles
- La descripción del tipo. A partir del código de tipo de producto, puedes extraer su descripción del fichero `types.bin`. El código de tipo de producto que aparece asociado a un producto en `products.bin` siempre existirá en el fichero `types.bin`. Es decir, no vamos a encontrar un producto cuyo código de tipo de producto no esté descrito en este fichero
- El precio del producto
- Número de unidades de producto en el almacén. Si el producto no es una novedad esta temporada, se exportarán como máximo 100 unidades

Si el nombre del producto, la descripción del producto o la descripción del tipo tienen espacios en blanco en su contenido, a la hora de guardarlos en el fichero de salida se deberán escribir entre comillas dobles (").

El programa puede recibir tres argumentos que indican cómo deberá proceder:

- `-f <nombre fichero>`: nombre del fichero de salida. Este argumento es obligatorio
- `-s <nombre separador>`: separador a utilizar entre cada campo del fichero de salida. Los valores aceptados son `pc` para punto y coma (;), `dp` para dos puntos (:) y `co` para coma (,). Este argumento es obligatorio
- `-v`: este argumento es opcional. Si se aporta, el programa mostrará por pantalla toda la información de los productos conforme los vaya leyendo de fichero (en el formato que tú decidas)

Estos argumentos pueden aparecer en cualquier orden. Ejemplos válidos de ejecución:

```
$ ej1 -f ficheroSalida.txt -s pc -v
```

```
$ ej1 -s dp -f ficheroSalida.txt
```

Si apareciera algún argumento incorrecto, o aparecieran argumentos repetidos, o faltara alguno de los argumentos obligatorios o el nombre de fichero, habría que mostrar un error de argumentos incorrectos y finalizar el programa.

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