

Fundamentos de la Programación

Curso 2017–2018

Examen de septiembre

Tiempo disponible: 3 horas

Se pide construir un programa modular en C++ que permita gestionar una juguetería de coleccionista. La solución constará de un programa principal (main.cpp) y tres módulos: Juguete, ListaJuguetes y Coleccion.

Módulo Juguete (1,5 puntos)

Declara un tipo de estructura `tJuguete` con información necesaria para representar el identificador de un juguete (`string`, sin espacios) que será único; la década en la que se construyó el juguete, que será un enumerado (años 50, años 60, años 70 y años 80); número de unidades (`long int`), y el precio base del juguete (número real).

Implementar como mínimo las siguientes operaciones:

- `precioVenta()`: calcula el precio de venta del juguete, teniendo en cuenta que los juguetes de los 50 valen un 20% más que su precio base, los de los 60 un 15% más, los de los 70 un 10% más y los de los 80 un 5% más.
- `mostrarJuguete()`: muestra la información de un juguete: su identificador, década, unidades, precio base y precio de venta. El precio base y el precio de venta se mostrarán con 2 decimales.

Módulo ListaJuguetes (4 puntos)

Declara un tipo `tListaJuguetes` para guardar un inventario de juguetes. Esta lista se implementará con un **array dinámico** y estará ordenada por el identificador del juguete. En esta lista, en vez de tener un contador con el número de elementos totales, tendrá 4 contadores (uno para los juguetes de los años 50, otro para los juguetes de los años 60, otro para los de los 70, y otro para los de los 80). Así, el número total de juguetes será la suma de los 4 contadores. Recuerda que los identificadores no estarán repetidos.

Implementa, al menos, las siguientes funciones:

- `cargar()`. Carga de fichero una lista de juguetes. En el fichero aparecerá primero el número de juguetes y después, en orden, la información de cada uno de ellos. La década aparecerá como número (50, 60, 70, 80), ver ejemplo. En el fichero la lista de juguetes no está necesariamente ordenada y puede contener juguetes repetidos (en tal caso se puede asumir que sólo podrán diferir en el número de unidades), pero una vez cargada, la lista inventario estará ordenada por

identificador y no contendrá repeticiones. **Es necesario insertar de forma ordenada, no es válido cargar y después ordenar.**

- `buscar()`. Dado el identificador de un juguete y una lista, devuelve un booleano que informa si el juguete con el identificador dado está en la lista. Además, devuelve un número que indica la posición del juguete (si está en la lista), o la posición que le correspondería en la lista (si no está en la lista).
- `insertar()`. Dado el identificador de un juguete, una década, un número de unidades y un precio base, inserta un juguete en la lista. Si el juguete no estaba en la lista (comprobar por identificador únicamente, no hace falta comprobar el resto de los campos) se inserta en la misma. Si el identificador ya existía en la lista, sólo hay que actualizar el número de unidades de ese juguete (sumando a las que ya había).
- `eliminar()`: Dado el identificador de un juguete y una lista, actualiza el número de unidades, eliminando una unidad del juguete correspondiente. Si tras la eliminación no quedan unidades de ese juguete, hay que eliminarlo de la lista. Ten en cuenta que tendrás que decrementar el contador correspondiente a la década del juguete que se está eliminando. Tras la eliminación es necesario desplazar los elementos del array, no deben quedar huecos.

La función devuelve `true` si, al eliminar una unidad, se ha eliminado el juguete (es decir, si ya no quedan más) y `false` en caso contrario.

- `mostrar()`: Muestra el inventario.
- `liberar()`. Libera la memoria dinámica reservada.

Módulo Coleccion (3,5 puntos) Máx. 50 juguetes

Declara un tipo `tColeccion` para guardar una colección concreta de juguetes. Esta lista se implementará con un array estático de punteros a variables dinámicas.

Implementa, al menos, las siguientes funciones:

- `crearColeccion()`. Dados una lista de juguetes, un precio máximo (base) para la colección y una década (50, 60, 70 u 80) genera una colección. Para añadir cada juguete, se elige secuencialmente de la lista de juguetes de entrada, siempre que el juguete sea de la década indicada. A la colección solamente se añade una unidad de cada juguete.

La generación termina cuando no queden juguetes de una década, o justo en el momento en el que el valor total de la colección exceda el precio máximo (el último juguete entrará en la colección, aceptando ese exceso de precio).

Los juguetes añadidos a la colección tienen que ser eliminados del inventario, llamando a la función `eliminar()` del módulo `ListaJuguetes`.

- `mostrar()`. Muestra la colección de juguetes. Debe ser implementado como una función recursiva.

Programa Principal (1 punto)

Carga el contenido del archivo `juguetes.txt` en la lista de juguetes y muestra al usuario un menú:

1. Visualizar inventario de juguetes.
2. Crear y mostrar una coleccion.
3. Salir.

Debe liberarse toda la memoria dinámica utilizada. El profesor te proporcionará un esquema del programa principal que has de rellenar con las llamadas a las funciones apropiadas en cada caso.

Entrega del examen:

- 1- Entrega el código del programa (sólo `.cpp` y `.h`) utilizando la herramienta de FTP del escritorio. Asegúrate de entregar una versión sin errores de compilación.
- 2- Añade al inicio de tus archivos un comentario con tus datos:

```
/* Apellidos: Nombre: DNI: Puesto: */
```
- 3- Abre la herramienta de entrega de exámenes por ftp que hay en el escritorio de tu ordenador.
- 4- Úsala para subir tus archivos (arrastra tus ficheros hacia la ventana derecha).
- 5- Pasa por el ordenador del profesor, pregúntale si tu archivo se ha recibido correctamente, y firma.

Para que se muestre la memoria no liberada, usa este comando:

```
_CrtSetDbgFlag(_CRTDBG_ALLOC_MEM_DF | _CRTDBG_LEAK_CHECK_DF);
```

E incluye este código:

```
// archivo checkML.h
#ifdef _DEBUG
#define _CRTDBG_MAP_ALLOC
#include <stdlib.h>
#include <crtdbg.h>
#ifndef DBG_NEW
#define DBG_NEW new ( _NORMAL_BLOCK , __FILE__ , __LINE__ )
#define new DBG_NEW
#endif
#endif
```

Ejemplo de ejecución:

1. Visualizar inventario de juguetes.
2. Crear y mostrar una coleccion.
3. Salir.

Elija una opcion: 1

Comba

decada 60

200 unidades

Precio base: 4.50, pr. venta: 5.17

GafasBroma

decada 80

100 unidades

Precio base: 15.68, pr. venta: 16.46

HeroQuest

decada 80

10 unidades

Precio base: 65.50, pr. venta: 68.78

Peonza

decada 50

5 unidades

Precio base: 6.75, pr. venta: 8.10

Pompas

decada 70

20 unidades

Precio base: 1.25, pr. venta: 1.38

RatonMadera

decada 50

15 unidades

Precio base: 25.50, pr. venta: 30.60

Taba

decada 50

50 unidades

Precio base: 14.45, pr. venta: 17.34

ZapatillasRaras

decada 70

10 unidades

Precio base: 80.00, pr. venta: 88.00

-
1. Visualizar inventario de juguetes.
 2. Crear y mostrar una coleccion.
 3. Salir.

Elija una opcion: 2

Decada (50, 60, 70, 80): 80

Precio maximo: 10

GafasBroma

decada 80

1 unidades

Precio base: 15.68, pr. venta: 16.46

-
1. Visualizar inventario de juguetes.
 2. Crear y mostrar una coleccion.
 3. Salir.

Elija una opcion: 1

Comba

decada 60

200 unidades

Precio base: 4.50, pr. venta: 5.17

GafasBroma

decada 80

99 unidades

Precio base: 15.68, pr. venta: 16.46

HeroQuest

decada 80

10 unidades

Precio base: 65.50, pr. venta: 68.78

Peonza

decada 50

5 unidades

Precio base: 6.75, pr. venta: 8.10

Pompas

decada 70

20 unidades

Precio base: 1.25, pr. venta: 1.38

RatonMadera

decada 50

15 unidades

Precio base: 25.50, pr. venta: 30.60

Taba

decada 50

50 unidades

Precio base: 14.45, pr. venta: 17.34

ZapatillasRaras

decada 70

10 unidades

Precio base: 80.00, pr. venta: 88.00

-
1. Visualizar inventario de juguetes.
 2. Crear y mostrar una coleccion.
 3. Salir.

Elija una opcion: 2

Decada (50, 60, 70, 80): 50

Precio maximo: 100

Peonza

decada 50

1 unidades

Precio base: 6.75, pr. venta: 8.10

RatonMadera

decada 50

1 unidades

Precio base: 25.50, pr. venta: 30.60

Taba

decada 50

1 unidades

Precio base: 14.45, pr. venta: 17.34

-
1. Visualizar inventario de juguetes.
 2. Crear y mostrar una colección.
 3. Salir.

Elija una opción: 3