

Examen final – 6 de junio de 2013

Tiempo disponible: 3 horas

Una empresa de informática quiere mantener las configuraciones de los equipos que venden a sus clientes. Se ha de mantener una lista de hasta 300 equipos **ordenada por código de equipo**.

Para cada equipo se mantiene la siguiente información (tipo `tEquipo`):

- Código del equipo (cadena sin espacios)
- Precio del equipo (suma del precio de los componentes más un 21% de IVA)
- Lista de los componentes del equipo (hasta 20 componentes) (**array dinámico**)

De cada componente se guarda la siguiente información (tipo `tComponente`):

- Código del componente (cadena sin espacios)
- Nombre del componente (cadena con espacios)
- Precio del componente (real)

La relación de componentes de los equipos está en un archivo de texto `equipos.txt` que nuestro programa deberá leer. Contiene una serie de líneas, cada una con estos datos separados por un espacio: código del equipo, código, precio y nombre del componente. Termina con una línea con "X" como código. El archivo será siempre correcto (no hay que comprobar posibles errores, más allá de que exista el archivo).

Ejemplo de archivo `equipos.txt`:

```
PC223 DSK32G6 86.45 Disco 500 Gb
EB1234 XK0T11 112.4 Placa base dual proc.
PC223 M00R23H 21.24 Ventilador Powervent
EB1234 DSK002G45 65.05 Disco 350 Gb
EB1234 003R4H 34 Unidad CD/DVD
...
X
```

Desarrolla un programa en C++ que incluya los siguientes subprogramas:

- ✓ `cargaEquipos()`: carga la información del archivo `equipos.txt`; para cada línea localiza el código de equipo en la lista (si no existe, lo inserta) y añade un componente en su lista de componentes con el código, precio y nombre; y actualiza el precio del equipo. Devuelve la lista con los datos leídos, o vacía si el archivo no existe. Usa las siguientes...

- ✓ `buscaEquipo()`: dada una lista ordenada de equipos y un código, devuelve la posición (índice) para ese código en la lista; y si se ha encontrado o no un equipo con ese código.
- ✓ `insertaEquipo()`: dada una lista de equipos y un equipo, añade éste a la lista. La lista debe seguir ordenada por código.

En ningún momento se debe hacer uso de algoritmo de ordenación alguno.

- ✓ `insertaComponente()`: dado un equipo y un componente, añade éste último a la lista de componentes del equipo y actualiza el precio del equipo.
- ✓ `muestraEquipo()`: muestra la información de un equipo con este formato:

```
EB1234
  XK0T11 Placa base dual proc. - 112.40 Eur
  DSK002G45 Disco 350 Gb - 65.05 Eur
  003R4H Unidad CD/DVD - 34.00 Eur
```

```
...
Precio del equipo: 342.56 Eur
```

- ✓ `muestraEquipos()`: muestra la lista de equipos.
- ✓ `actualiza()`: aplica a todos los componentes el porcentaje de incremento que se proporciona, actualizando los precios de los equipos.

El programa principal comenzará cargando la información del archivo en la lista de equipos. A continuación mostrará la lista de equipos. A continuación actualizará el precio de los componentes, para terminar mostrando de nuevo la lista de equipos.

Se valorará la estructuración del código, el uso adecuado de los esquemas de recorrido y búsqueda, la comunicación entre subprogramas, ...

Puntuación:

Estructuras de datos: 1,5 puntos
`cargaEquipos()`: 1,75 puntos
`buscaEquipo()`: 0,75 punto
`insertaEquipo()`: 0,75 punto
`insertaComponente()`: 0,5 puntos
`muestraEquipo()`: 0,75 punto
`muestraEquipos()`: 0,25 punto
`actualiza()`: 1,25 puntos
Programa principal: 1 punto
Programa modular: 1 punto (*entrega un ZIP con todos los archivos de código*)
Estilo y comentarios: 0,5 puntos

Deberás entregar el código del programa a través del Campus Virtual. Asegúrate de entregar una versión sin errores de compilación.