

Convocatoria extraordinaria – 10 de septiembre de 2014

Tiempo disponible: 3 horas

Se pide construir un programa modular que muestre notas de exámenes agrupadas por alumnos. Habrá, al menos, cuatro módulos: *Nota*, *ListaExámenes*, *ListaAlumnos* y principal (*main.cpp*).

Módulo *Examen* (0,5 puntos)

Declara un tipo de estructura `tExamen` con tres campos: fecha (AAAA/MM/DD), nota numérica y nota cualitativa (SS, AP, B, NT o SB). Implementa, al menos, las siguientes funciones:

- ✓ `muestra`: Dado un examen muestra su información con el siguiente formato:
`2014/09/01 - 9.0 - SB`
- ✓ `notaCualitativa`: Dada una nota numérica, devuelve su calificación cualitativa (<5: SS; >= 5 y < 6: AP; >=6 y < 7: B; >=7 y < 9: NT; >= 9 y <=10: SB).

Módulo *ListaExámenes* (2 puntos)

Máx. 50 exámenes

Declara un tipo `tListaExámenes` para listas de exámenes de longitud variable **implementadas con array dinámico** (sin orden). Implementa, al menos, las siguientes funciones:

- ✓ `estadisticaNotas`: Dada una lista de exámenes devuelve un **array estático** de cinco posiciones donde cada posición representa el número de suspensos, aprobados, bienes, notables y sobresalientes de la lista de exámenes.
- ✓ `muestraLista`: Dada una lista de exámenes y una posición muestra en pantalla los exámenes a partir de esa posición. Se implementará de forma recursiva. Ejemplo:
`2014/09/01 - 7.0 - NT`
`2014/09/03 - 9.0 - SB`
- ✓ `muestra`: Muestra en pantalla una lista de exámenes junto a sus estadísticas:
`...`
`2014/09/01 - 7.0 - NT`
`2014/09/03 - 9.0 - SB`
`SS: 4 AP: 2 B: 0 NT: 2 SB: 1`
- ✓ `insertaNota`: Dada una lista de exámenes y un examen añade el examen al final de la lista. Si la lista está llena, se ignora el nuevo examen.
- ✓ `destruye`: Dada una lista de exámenes libera la memoria dinámica que usa.

Módulo *ListaAlumnos* (4,5 puntos)

Máx. 100 alumnos

Declara un tipo `tAlumno` que guarde su NIF y su lista de exámenes. Declara también un tipo de estructura `tListaAlumnos` para listas de alumnos de longitud variable **implementadas**

con array estático de datos dinámicos. La lista de alumnos se debe mantener **ordenada por NIF**. Implementa, al menos, los siguientes subprogramas:

- ✓ **totalEstadisticas:** Dada una lista de alumnos, devuelve una array estático de cinco posiciones donde cada posición representa respectivamente el número de suspensos, aprobados, bienes, notables y sobresalientes de todos los exámenes existentes.
- ✓ **muestra:** Dada una lista de alumnos muestra en la pantalla los exámenes de cada alumno y sus estadísticas, así como al final las estadísticas totales. Por ejemplo:

```
-----  
Alumno: 11111111A  
2014/09/02 - 5.5 - AP  
...  
SS: 0 AP: 1 B: 1 NT: 5 SB: 2  
-----  
Alumno: 22222222B  
...  
2014/09/01 - 7.0 - NT  
2014/09/03 - 9.0 - SB  
SS: 4 AP: 2 B: 0 NT: 2 SB: 1  
-----  
...  
Total.- SS: 112 AP: 215 NT: 43 SB: 21 MH: 3
```

Entrega el código del programa
a través del Campus Virtual
(sólo .cpp y .h, comprimidos en un ZIP)
*¡Asegúrate de entregar una versión
sin errores de compilación!*

- ✓ **busca:** Dada una lista de alumnos y un NIF, localiza en la lista un alumno con ese NIF. Si lo encuentra devuelve la posición que ocupa, si no lo encuentra devuelve la posición que debería de ocupar. La búsqueda se hará iterativa y de la forma más eficiente.
- ✓ **insertaAlumno:** Dada una lista de alumnos, un NIF y un examen, y una posición, inserta un nuevo alumno con ese NIF (y ese examen como primer examen) en la lista de alumnos en la posición dada. Si la lista está llena, se ignora el nuevo alumno.
- ✓ **carga:** Carga en una lista de alumnos los exámenes del archivo `datos.txt`, que contiene en cada línea la información de un examen: NIF, fecha, y nota numérica. El archivo termina con `XXX` como NIF. Ejemplo de archivo `datos.txt`:

```
22222222B 2014/09/05 8.5  
11111111A 2014/09/03 5.5  
44444444C 2014/09/09 9  
...  
XXX
```

Se lee cada examen y se localiza el alumno. Si no existe, se añade a la lista de alumnos (manteniendo el orden de NIFs). Si existe, se añade el examen a la lista del alumno.

- ✓ **destruye:** Dada una lista de alumnos libera la memoria dinámica que utiliza.

Módulo principal (0,5 puntos)

Carga la información del archivo `datos.txt` en una lista de alumnos, muestra la información cargada y libera la memoria dinámica utilizada.

Se valorará la legibilidad, así como el uso adecuado de los esquemas de recorrido y búsqueda, de la comunicación entre subprogramas y de la memoria (2,5 puntos).