

Examen final – 20 de junio de 2011

Tiempo disponible: 2 horas

**Ejercicio único** (8 puntos)

Se pide construir un programa (**junio.cpp**) que gestione una lista de apuestas.

Programa principal (1 punto)

Comenzará cargando la lista con la información del archivo. Si el archivo no existe, lo indicará y terminará la ejecución. Si la carga tiene éxito, ordenará la lista como se indica más adelante y ofrecerá a continuación un menú al usuario:

- 1 - Mostrar la lista de apuestas
- 2 - Buscar una apuesta
- 0 - Salir

El menú se repetirá hasta que el usuario elija la opción 0.

Estructuras de datos (1,5 puntos)

Debes declarar las estructuras de datos adecuadas para la información que se va a obtener del archivo de entrada (*apartado siguiente*).

Carga de la lista (1,5 puntos)

Se parte de un archivo de texto que se proporciona (**datos.txt**) que contiene una serie de líneas, cada una con la información de un registro de la lista y estando los datos separados por un espacio: id (entero sin signo), nº de números en la apuesta (entero sin signo de 6 a 12), número1 (de 1 a 49), número2 (de 1 a 49), ..., precio (real), nº de aciertos (entero sin signo), premio (real) y cliente (una cadena de hasta 30 caracteres, sin espacios). Habrá 500 líneas como máximo. La última línea tendrá sólo un 0 indicando que no hay más registros. Ejemplo de archivo de entrada:

```
121 6 1 4 23 12 48 33 1.0 2 0.0 Soler
132 7 3 42 31 7 24 49 12 7.0 3 16.0 Vegas
118 8 5 2 19 28 43 31 15 16 56.0 4 2317.5 Santana
216 6 33 11 22 44 6 30 1.0 0 0.0 Jovellanos
333 7 42 37 26 8 17 30 49 7.0 5 114654.0 Rosillo
0
```

Se creará la lista de apuestas a partir de la información del archivo.

### Ordenación de la lista (1,5 puntos)

Ordenará la lista de mayor a menor ID. La ordenación de la lista se realizará por el método de selección directa y se implementará de forma recursiva.

El subprograma de ordenación recibirá la lista y la posición del array que se procesa en cada llamada (0, 1, 2, ...):

```
void ordenar(tLista &lista, int pos);
```

El subprograma localizará el registro con mayor ID de lo que queda de lista (desde pos) e intercambiará el de esa posición con el de mayor ID para a continuación realizar una llamada recursiva para que se ordene lo que resta de lista de apuestas.

Los números de cada apuesta no hay que ordenarlos.

### 1 - Mostrar la lista (1 punto)

Mostrará la lista de apuestas con el siguiente formato de ejemplo para cada una:

```
333 Rosillo (7.00 Euros)
    42 37 26 8 17 30 49 --> 5 aciertos (114654.00 Euros)
```

### 2 - Localizar una apuesta (1,5 puntos)

Pedirá un ID y localizará el registro con ese ID (si existe). Lo devolverá a través de un puntero a una variable dinámica (que se creará antes de llamar al subprograma de búsqueda). Si se ha encontrado, mostrará su información con el mismo formato del apartado anterior.

Cuida el estilo, define todos los tipos necesarios y estructura bien el código.

Una vez que hayas terminado el ejercicio, *súbelo* al Campus Virtual:  
Tarea identificada como **Examen de junio: Ejercicio**

*¡No olvides poner al principio del programa un comentario con tu nombre!*