

# Estructura de Datos y Algoritmos

Grado de Desarrollo de Videojuegos. Curso 2022-2023  
Examen final. Convocatoria ordinaria      Tiempo: 3 horas

## Instrucciones

- La entrega se realiza en el juez automático de los laboratorios accesible desde la url <http://exacrc> (cada ejercicio en su correspondiente problema del juez, acabados respectivamente en Ej1, Ej2 y Ej3). Para acceder debes usar el usuario/contraseña que has recibido al comienzo del examen.
- Al principio de cada fichero .cpp debe aparecer, en un comentario, vuestro nombre y apellidos, dni y puesto de laboratorio. También debéis incluir unas líneas explicando qué habéis conseguido hacer y qué no.
- Todo lo que no sea código C++ (explicaciones, respuestas a preguntas, etc.) debe ir en los propios ficheros en comentarios debidamente indicados.
- Los TADs, las plantillas y ficheros de entradas de ejemplo para cada ejercicio se descargan desde <http://exacrc/EDA-Enero23.zip>.
- Podéis realizar varias entregas para un mismo ejercicio pero solamente se tendrá en cuenta la última.
- Podéis acceder a la referencia de C++ en <http://exacrc/cppreference>

## Ejercicio 3 [4 puntos]

Nos piden un programa que produzca la clasificación final de un concurso de programación conociendo los envíos de soluciones a los distintos problemas que han realizado los equipos participantes. Para la clasificación se tienen en cuenta las siguientes reglas:

- El criterio principal para ordenar a los equipos es el número de problemas resueltos. Cuantos más problemas se resuelvan, mejor será el puesto en la clasificación.
- A igualdad de problemas resueltos, los equipos que hayan tardado menos tiempo se clasifican primero.
- El tiempo utilizado por un equipo es igual a la suma del tiempo necesario para resolver cada problema resuelto correctamente.
- El tiempo necesario para resolver un problema es la suma del tiempo transcurrido desde el comienzo del concurso hasta el primer envío correcto a ese problema, más 20 minutos de penalización por cada envío (a ese problema) incorrecto anterior. Los envíos posteriores al primer envío correcto se ignoran.
- Los problemas que finalmente no se resuelven no penalizan (aunque se hayan hecho envíos incorrectos).
- Si dos equipos resuelven el mismo número de problemas en el mismo tiempo, serán ordenados por su nombre de menor a mayor.

En concreto, se pide implementar la función `procesarEnvios` la cual lee de teclado la información de todos los envíos y devuelve (mediante parámetro por referencia) una estructura con la salida, de manera que ésta simplemente deberá ser recorrida para imprimir la salida en el formato pedido. Se valorarán de manera notable el uso adecuado de estructuras de datos auxiliares, la eficiencia y la complejidad de la implementación, la cual debes indicar y justificar. Puedes usar si lo necesitas la función `std::sort` del paquete `algorithm` de la STL cuya complejidad puedes asumir que es  $O(n * \log(n))$ .

**Entrada** La entrada consta de una serie de casos de prueba. En la primera línea aparece el número de casos que vendrán a continuación. Cada caso consiste en una serie de descripciones de envíos, cada uno en una línea. Cada envío está formado por el nombre del equipo (una cadena de caracteres sin espacios), el nombre del problema (otra cadena de caracteres sin espacios), los minutos transcurridos desde el comienzo del concurso, y el veredicto del juez (AC, WA, TLE, etc). Un envío se considera correcto solamente si el veredicto es AC. Los envíos de cada caso están ordenados de menor a mayor número de minutos. El último envío está seguido de una línea con la palabra FIN.

**Salida** Para cada caso de prueba se escribirá la clasificación final. Ésta contendrá una línea por cada equipo que haya realizado algún envío, donde aparecerá el nombre del equipo, el número de problemas resueltos y el tiempo empleado. Los equipos aparecen ordenados según las reglas de clasificación, comenzado por el mejor clasificado. La clasificación de cada caso estará seguida por una línea con "---".

**Ejemplos de Entrada/Salida:**

<pre> 2 LosIntrepidos PequalsNP 8 TLE EquipoA ElSolitario 30 AC JavaForever ElSolitario 30 RTE LosIntrepidos PequalsNP 100 AC LosIntrepidos Sumar2y2 101 AC Lentorros Sumar2y2 200 AC LosIntrepidos ElSolitario 210 WA B_team Horarios 220 WA FIN NoDoyNiUna A 10 TLE NoDoyNiUna B 12 RTE NoDoyNiUna C 16 WA FIN </pre>	<pre> LosIntrepidos 2 221 EquipoA 1 30 Lentorros 1 200 B_team 0 0 JavaForever 0 0 --- NoDoyNiUna 0 0 --- </pre>
---	---