



# Trabajo Práctico Tópicos de Programación (Martes tarde)

Como desarrollador de ansi C se pide crear un programa para mergear (unir) múltiples archivos de alumnos.

Se entregan junto con el enunciado varios archivos binarios (en este caso son 5 pero podrían ser más o menos) conformados por estudiantes.

La estructura estudiantes está compuesta por los siguientes campos:

<u>Campo</u>	<u>Tipo de dato</u>
dni	unsigned long long
fechaDeInscripcion	unsigned long long
nombreYApellido	char[30]
promedio	float

Consideraciones respecto de los archivos:

- Cada archivo internamente está ordenado por fecha de inscripción.
- No se asegura el ordenamiento entre los distintos archivos entregados.
- También se asegura que la fecha de los archivos no se solapan.
- Cada archivo contiene como máximo 10 registros.
- Por limitaciones del sistema dónde se quiere correr el programa final de unión sólo puede tener 12 registros en memoria como máximo.



**UNLaM**  
Universidad  
Nacional de  
La Matanza



**Tópicos de programación  
3635**

**Comisión: 03-2600**



**Se pide:**

1. Cree una función de ordenamiento y merge genérica (Es decir que admita estudiantes pero que pueda admitir otras estructuras).
2. Cree un único **archivo binario** con el listado completo de alumnos ordenado por el campo Fecha de Inscripción (Utilizando la función del punto anterior).
  - a. Normalice los nombres y apellidos poniendo la primera letra del nombre y del apellido en mayúscula.
3. Muestre los datos de los 5 alumnos con mejor promedio (ordenados por el promedio).
4. Guarde el resultado anterior en un **archivo de texto** llamado "MejoresAlumnos.csv" (separados por punto y coma) mostrando la fecha en un formato legible para los humanos.

 <b>UNLaM</b> Universidad Nacional de La Matanza	<b>Tópicos de programación 3635</b>  <b>Comisión: 03-2600</b>	 <b>DIIT</b> Departamento de Ingeniería e Investigaciones Tecnológicas
-------------------------------------------------------------------------------------------------------------------------------------------	-------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------

### Formato de Entrega:

- Todos los integrantes deben subir su TP en un .zip con el formato:
  - DNI\_APELLIDO\_NOMBRE\_TP.zip
- Deberán borrar las carpetas “bin” y “obj”.
- Dentro del zip deben estar el proyecto con el “main” y las librerías que hayan creado.
- El TP deberá ser entregado por la plataforma MleL.

### Condiciones Generales:

- Grupos de entre 2 y 4 integrantes.
- Fecha de entrega: 25/06/2024.
- El mismo se defenderá el 25/06/2024.
- El TP no tiene reentrega.

### Lea con atención las condiciones de aprobación:

- Cumplimentar con la consigna completa.
- 0 errores 0 warnings.
- Que funcione mínimo para todos los casos de prueba que presentan (agregue al menos 1).
- Vectores y cadenas de texto deberán ser manipulados utilizando aritmética de punteros.
- Las soluciones tienen que ser eficientes:
  - En el uso de memoria, por tanto, no declare vectores o matrices auxiliares si no es necesario.
  - En el uso de archivos temporales o auxiliares.
  - En cantidad de ciclos de procesador y en el caso de matrices las soluciones deben ser óptimas.
  - Los recorridos de archivos tienen que ser mínimos.
- Nunca acceda a memoria que no le pertenece.
- No deje archivos abiertos.
- Declare variables al inicio del bloque y no utilice VLA (Variable length arrays).
- Se evaluará la prolijidad general:
  - Código prolijo, dividido en funciones,
  - Nombres descriptivos,
  - Nombres significativos de variables.
- Las soluciones tienen que estar desarrolladas en ANSI C para garantizar tanto como sea posible compatibilidad multiplataforma.
- Nunca mezcle funciones de manipulación de archivos de texto con funciones de operación de archivos binarios.