

 UCA <small>PONTIFICIA UNIVERSIDAD CATÓLICA ARGENTINA</small> Facultad de Ingeniería y Ciencias Agrarias	Calificación		21 / 12/ 2017	FINAL	TEMA 1
	LEGAJO:		INFORMÁTICA 2 - PROGRAMACIÓN ESTRUCTURADA		Hojas:
	APELLIDO:		NOMBRE:		
	Para aprobar PARTE III deberá contar con al menos un ejercicio BIEN (de dos).				[PARTE III DE III]

PARTE III - Programación (38 puntos):

3 - Programación

3.1) - Desarrollar la función **unificar** que recibe como parámetro dos string (char*) con el nombre de un archivo CVS (en cada parámetro). Los dos archivos contienen información de los alumnos que pertenecen a dos comisiones de Física (*una comisión por archivo*) con la siguiente estructura de datos:{ *nro_registro*, *nombre*, *apellido*}; y los datos están ordenados por número *nro_registro*. La función deberá unificar las dos comisiones en un sólo archivo (*nuevo archivo*) cuyos datos también deberán estar ordenados por *nro_registro*. Considerar la posibilidad que un alumno se encuentre inscripto en ambas comisiones por error en cuyo caso se incluirá en el nuevo archivo una sola vez.

Archivo	Estructura de datos	Ejemplo de contenido
comisionAM.CSV	nro_registro (<i>int</i>) nombre (<i>char*</i>) apellido (<i>char*</i>)	170022,Juan,Rodriguez 170050,Maria,Perez 170125,Lorena,Ledesma 170245,Tomas,Garcia
comisionBM.CSV	nro_registro (<i>int</i>) nombre (<i>char*</i>) apellido (<i>char*</i>)	170012,Luciana,Mendez 170125,Lorena,Ledesma 170179,Ignacio,Ponzi 170321,Pedro,Martin 170450,Alberto,Alonso
comisionABM.CSV (nuevo archivo)	nro_registro (<i>int</i>) nombre (<i>char*</i>) apellido (<i>char*</i>)	170012,Luciana,Mendez 170022,Juan,Rodriguez 170050,Maria,Perez 170125,Lorena,Ledesma 170179,Ignacio,Ponzi 170245,Tomas,Garcia 170321,Pedro,Martin 170450,Alberto,Alonso

3.2) - Desarrollar la función **listado** que recibe por parámetro un lista del tipo **t_alu**, con un campo con el apellido de cada alumno (*qué no se repite en la lista*) y otro campo con una lista del tipo **t_nota**, que contiene las notas de parciales rendidos por el alumno. La función deberá confeccionar e imprimir un listado indicando el apellido de cada alumno con su promedio de parciales ordenado, descendentemente por el promedio.

<pre>struct s_alu{ char* apellido; t_nota lstNota; struct s_alu* sig; }; typedef struct s_alu* t_alu; struct s_nota{ int valor; struct s_nota* sig; }; typedef struct s_nota* t_nota;</pre>	<div>Ejemplo de una lista con datos:</div> <div>lista=</div> <div><pre>{ "Garcia", {5,4,2}, "Lopez", {10,6,7,2,6}, "Cardenas", {4,6,7,8,9}, "Molinari", {2,6,4}, "Estevez", {9,8,7,8} }</pre></div>	<div>Ejemplo de salida de un listado impreso:</div> <table><tr><th>Apellido</th><th>Promedio</th></tr><tr><td colspan="2">-----</td></tr><tr><td>Estevez</td><td>8.00</td></tr><tr><td>Cardenas</td><td>6.80</td></tr><tr><td>Lopez</td><td>6.20</td></tr><tr><td>Molinari</td><td>4.00</td></tr><tr><td>Garcia</td><td>3.67</td></tr></table>	Apellido	Promedio	-----		Estevez	8.00	Cardenas	6.80	Lopez	6.20	Molinari	4.00	Garcia	3.67
Apellido	Promedio															

Estevez	8.00															
Cardenas	6.80															
Lopez	6.20															
Molinari	4.00															
Garcia	3.67															

- Algunos criterios mandatorios para aprobar los problemas:
- Declarar todos los "struct" y "typedef" que utilice para resolver los ejercicios.
 - No desperdiciar memoria. Liberar todos los espacios que se hayan reservado y que no serán utilizados.
 - Se pueden programar funciones adicionales, respetando los puntos anteriores y declarando sus parámetros y retorno. Considere que las estructuras recibidas como parámetro por las funciones pueden estar vacías (nulas).

El examen está compuesto por tres partes: Completar Programa, opción múltiple y programación. Para aprobar el examen se requiere tener aprobado las tres partes y además **sumar un total de 65 puntos**, sin excepción.