



Apellido y Nombres	Legajo	# de Hojas	Profesor

### **Normas Generales**

Numere las hojas entregadas. **Lea detenidamente cada pregunta y consulte las dudas de interpretación que pudieran surgir.** Complete en la primera hoja la cantidad total de hojas entregadas. Debe identificarse cada hoja con: Nombre, Apellido, Legajo, y Nombre del Profesor del curso. **Por favor entregar esta hoja y las restantes del tema junto al examen.**

### **Condición de aprobación examen**

*Sección teórica:* 2 de los 3 puntos

*Sección práctica:* ejercicio 1 sin errores conceptuales y el ejercicio 2 o el ejercicio 3

Se considerara mal cualquier ejercicio en el que se genere una violación de segmento sin importar el resto de la resolución.

Se considerara mal cualquier ejercicio que rompa la lista simple enlazada, dejando nodos sueltos o deje memoria sin liberar que deba ser liberada. No está permitido imprimir por stdout.

**Tiempo: 2 horas**

### **Sección Teórica**

- 1) ¿Cuál es la diferencia entre un header y una biblioteca? ¿Qué contiene cada uno?
- 2) Seleccione una de las siguientes opciones. Justifique su respuesta. ¿Qué factores condicionan el tamaño de un puntero?  

sizeof(int *)	>	sizeof(char *)
sizeof(int *)	=	sizeof(char *)
sizeof(int *)	<	sizeof(char *)
- 3) ¿Qué es un proceso? ¿Qué tipos de procesos existen? Describa brevemente los mecanismos de comunicación entre procesos.

### **Sección Práctica**

- 1) Se pide escribir una función llamada CargarDatos que lea la información de un archivo y genere una lista simplemente enlazada con la información del mismo. La lista debe estar ordenada por legajo en forma ascendente. El archivo contiene la información de una materia y posee el siguiente formato:

```
codigoMateria,legajoEstudiante,notaParcial\n
```

Si el legajo ya se encuentra dentro de la lista debe sumarse con la nota anterior e incrementar la cantidad de notas en uno (esto permitirá luego obtener el promedio del estudiante). El campo materia no será tenido en cuenta dentro la lista. El nodo y el prototipo de la función a implementar.

```
struct nodo_Alumno
{
    int legajo;
    int notaParcial;
    int cantNotas;
    struct nodo_Alumno *sig;
};
int CargarDatos (struct nodo_Alumno **lista, char *nombreArchivo);
```

La función **CargarDatos** devuelve:

- 0 , en el caso que el archivo no contenga datos
- 1 , en caso de error
- N , correspondiente a la cantidad de registros leídos.



Apellido y Nombres	Legajo	# de Hojas	Profesor

- 2) Desarrolle una función llamada `CalcularPromedio` que tome la lista creada anteriormente y calcule el promedio general.

El prototipo de la función a implementar es el siguiente:

```
int CalcularPromedio (struct nodo_Alumno *lista, float *promedio) ;
```

La función `CalcularPromedio` devuelve:

0 , en el caso que la lista no contenga datos  
N , correspondiente a la cantidad de estudiantes

- 3) Realice un programa que utilice correctamente la función `CargarDatos` y `CalcularPromedio`. Dicho programa recibe por argumentos del main diferentes archivos correspondientes a las materias donde se han volcado los datos de los estudiantes. El programa debe contemplar el control de errores y que el usuario que ejecute el programa pueda pasar la cantidad de archivos que necesite dependiendo el contexto.

A continuación se brinda un ejemplo a modo ilustrativo:

Ejemplo de uso : `./promedio archivoMateria1 archivoMateria2 archivoMateria3`

Se valorará en el ejercicio que proponga como manejar diferentes argumentos.