

Universidad Tecnológica Nacional											
Facultad Regional Avellaneda											
Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos											
Materia: <b>Laboratorio de Computación I</b>											
Apellido:					Fecha:						
Nombre:					Docente <sup>(2)</sup> :						
División:					Nota <sup>(2)</sup> :						
Legajo:					Firma <sup>(2)</sup> :						
Instancia <sup>(1)</sup> :	PP		RPP		SP		RSP		FIN	X	

(1) Las instancias validas son: 1<sup>er</sup> Parcial (**PP**), Recuperatorio 1<sup>er</sup> Parcial (**RPP**), 2<sup>do</sup> Parcial (**SP**), Recuperatorio 2<sup>do</sup> Parcial (**RSP**), Final (**FIN**) . Marque con una cruz.

(2) Campos a ser completados por el docente.

### Ejercicio 1

Se posee un array de estructuras del tipo "Persona":

```
struct S_Persona{
    char nombre[20];
    int edad;
};
typedef struct S_Persona Persona;
```

Escribir una función que ordene las estructuras en el array por edad (descendente) y por nombre (ascendente). El prototipo de la función es el siguiente:

```
void persona_ordenarPorEdadMayor(Persona* personas, int len)
```

Se pasa como argumento el array de personas y su longitud.

### Ejercicio 2

Dada la siguiente llamada a función:

```
if(buscarX(letras))
    printf("Hay por lo menos una letra X");
```

- Realizar el prototipo.
- Realizar el desarrollo, teniendo en cuenta que el vector "letras" tiene una extensión de 50 caracteres. La función deberá retornar un valor en caso de encontrar una letra X y otro valor en caso de que no haya coincidencia alguna.

### Ejercicio 3

Declarar un puntero a int llamado ptrInt, asignar de manera dinámica espacio de memoria para guardar cinco valores, cargándolo secuencialmente. Luego, redimensionarlo a diez cargando los cinco valores restantes. Mostrar los datos cargados y liberar recursos de memoria utilizados.

### Ejercicio 4

Crear una biblioteca (archivo Auto.c y Auto.h). Esta biblioteca contendrá funciones para interactuar con un tipo de dato llamado Auto que proveerá la propia biblioteca.

Los datos del Auto serán:

- Marca (char[20])
- modelo (int)
- color (int)
- patente(char[8])

La biblioteca deberá proveer "defines" para los colores (ROJO, VERDE, AZUL, GRIS, NEGRO, BLANCO)

Las funciones que deberá proporcionar la biblioteca son las siguientes:

- `int auto_setMarca(Auto* pAuto, char* marca)`

Carga el campo marca de la referencia de Auto pasada como argumento, con la marca pasada como argumento, validando que la marca posea más de 3 letras.

- `int auto_setModelo(Auto* pAuto, int modelo)`

Carga el campo modelo de la referencia de Auto pasada como argumento, con el modelo pasado, validando que sea un año entre 1970 y 2015

- `int auto_setColor(Auto* pAuto, int color)`

Carga el campo color de la referencia de Auto pasada como argumento, con el color pasado, validando que sea alguno de los colores proporcionados por los defines.

- `int auto_setPatente(Auto* pAuto, char* patente)`

Carga el campo patente de la referencia de Auto pasada como argumento, con la patente pasada como argumento, validando que la patente posea 3 letras un espacio y 3 números.

- `int auto_cargarAuto(Auto* pAuto, char* marca, int modelo, int color, char* patente)`

Internamente cargará la referencia de Auto pasada como argumento, con los valores pasados por argumento. Las validaciones de cada argumento deben ser las explicadas anteriormente

Todas las funciones devuelven 0 en caso de error y 1 en caso de éxito.

Realizar un pequeño programa para probar todas las funciones mencionadas.