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

(1) Las instancias validas son: 1er Parcial (**PP**), Recuperatorio 1er Parcial (**RPP**), 2do Parcial (**SP**), Recuperatorio 2do 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 nombre de manera ascendente (a-z).
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

Disponemos de un array de enteros que ya se encuentra cargado con valores. Generar las líneas de código necesarias para buscar el máximo valor. Se deberá tener en cuenta que el valor máximo puede repetirse. Mostrar el/los índice/s de el/los máximo/s encontrado/s.

Ejercicio 3

¿Qué diferencia existe entre BREAK y CONTINUE? Dar un ejemplo.

Ejercicio 4

Crear una biblioteca (archivos 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 como argumento, 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 como argumento, 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, 1 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.

*void auto_print(Auto *pAuto)*
Imprime por pantalla los datos del Auto.

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

Escribir un programa de test para probar las funciones.