

Universidad Tecnológica Nacional Facultad Regional Avellaneda						 UTN Fra				
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: 1^{er} Parcial (**PP**), Recuperatorio 1^{er} Parcial (**RPP**), 2^{do} Parcial (**SP**), Recuperatorio 2^{do} 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 según la edad, colocando a la edad mayor primero. 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 mínimo valor. Se deberá tener en cuenta que el valor mínimo puede repetirse. Mostrar el índice de el/los mínimos encontrados.

Ejercicio 3

¿En una función a que se denomina pasaje por valor y pasaje por referencia?

Ejercicio 4

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

Los datos del Socio serán:

- nombre (char[20])
- fechaNac (long int) en este formato AAAAMMDD
- tipo (int)

La biblioteca deberá proveer defines para los tipos: (ACTIVO, CADETE, INFANTIL, VITALICIO)

Las funciones que deberá proporcionar la biblioteca son las siguientes:

```
int socio_setNombre(Socio *pSocio, char *nombre)
```

Carga el campo nombre de la referencia de Socio pasada como argumento, con el nombre pasado como argumento, validando que el nombre posea más de 3 letras.

```
int socio_setFechaNac(Socio *pSocio, long int fechaNac)
```

Carga el campo FechaNac de la referencia de Socio pasada como argumento, con la FechaNac pasada como argumento, validando que sea entre 19000101 y 20150630.

*int socio_setTipo(Socio *pSocio, int tipo)*

Carga el campo tipo de la referencia de Socio pasada como argumento, con el tipo pasado como argumento, validando que sea alguno de los tipos proporcionados por los defines.

*int socio_cargarSocio(Socio *pSocio, char *nombre , long int fechaNac, int tipo)*

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

*void socio_print(Socio *pSocio)*

Imprime por pantalla los datos del socio.

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

Escribir un programa de test para probar las funciones.