

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: 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

```
char *s
s = "hola mundo";
```

¿qué valor se almacena en "s"?:

- 1) El 1er caracter de la cadena.
- 2) El string completo.
- 3) La dirección donde se almacena la cadena de caracteres.
- 4) Se lanza un error.

Ejercicio 3

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.

Ejercicio 4

```
void imprimirValores1(int* pValores)
{
    int i;
    for(i=0; i<5; i++)
        printf("%d\n",pValores[i]);
}

void imprimirValores2(int pValores[])
{
    int i;
    for(i=0; i<5; i++)
        printf("%d\n",*(pValores+i));
}

int valores[5];
int i;
valores[0]=0;
valores[1]=1;
valores[2]=2;
valores[3]=3;
valores[4]=4;

imprimirValores1(valores);
imprimirValores2(valores);
```

¿Cómo son las salidas por pantalla de cada función?. Justificar

- 1) Diferentes.
- 2) Iguales.
- 3) La función 2 no compila.
- 4) La función 1 lanza un error en tiempo de ejecución.