

Universidad Tecnológica Nacional

Facultad Regional Avellaneda



Técnico Superior en Programación - Técnico Superior en Sistemas Informáticos

**Materia: Laboratorio de Computación I**

|                            |           |                          |            |
|----------------------------|-----------|--------------------------|------------|
| Apellido:                  |           | Fecha:                   |            |
| Nombre:                    |           | Docente <sup>(2)</sup> : |            |
| División:                  |           | Nota <sup>(2)</sup> :    |            |
| Legajo:                    |           | Firma <sup>(2)</sup> :   |            |
| Instancia <sup>(1)</sup> : | <b>PP</b> | <b>RPP</b>               | <b>SP</b>  |
|                            |           |                          | <b>RSP</b> |
|                            |           |                          | <b>FIN</b> |
|                            |           |                          | <b>X</b>   |

(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

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 2

¿Qué diferencia existe entre BREAK y CONTINUE? Ejemplificar.

### Ejercicio 3

A partir de esta declaración:

```
int num;
```

Declarar un puntero a entero llamado **puntero**, hacerlo apuntar a **num** y cargar **num** con el valor 5 a través del puntero.

#### Ejercicio 4

Crear una biblioteca (archivo ListaProductos.c y ListaProductos.h) para el manejo de una lista de 100 productos.

Dentro del archivo se deberá definir la estructura "Producto" con los campos "codigo", "precio" y "flagLibre" (indica que el lugar no esta cargado). Y un array de este tipo con 100 posiciones.

Deberá tener las siguientes funciones:

```
void lista_borrarLista(void)
```

Itera la lista de productos y carga el flagLibre en 1.

```
Producto* lista_getProducto(int indice)
```

Devuelve un puntero a un tipo de dato Producto que apunta al item de la lista de productos indicada por el indice pasado como argumento. Si el indice esta fuera de rango o si el item esta libre, se devolverá un 0 (puntero a null)

```
int lista_setProducto(Producto* pProducto,int indice)
```

Se escribirán los datos del producto pasado por referencia en la posición indicada, si el indice esta fuera de rango se devolverá 0 de la contrario, se devolverá 1. Si el producto ya existe, también se devolverá un 0.

```
Producto* lista_getMasCaro(void)
```

Devuelve el producto mas caro de la lista, si la lista esta vacía, devuelve 0 (puntero a null)

```
int lista_borrar(Producto* pProducto)
```

Busca el producto pasado como argumento, y lo elimina de la lista. Devuelve 1 si lo elimino y 0 si el producto no estaba en la lista.

Realizar un pequeño programa para probar todas las funciones mencionadas.