

## Práctica 1: Punteros en C

### Punteros

- Escribir una función `rotate_values` que dados tres enteros `a`, `b` y `c` rote sus valores de manera tal que después de ejecutar la función en `a` se almacene el valor de `b`, en `b` el de `c` y en `c` el de `a`. Aclaración: la expresión “dados tres enteros” no significa que la función toma tres parámetros de tipo `int`.

- \_\_Escribir una función que dados un arreglo de enteros y su longitud, devuelva:

- \_\_el mayor elemento
- \_\_el menor elemento
- \_\_la posición donde ocurre el menor por primera vez
- \_\_la posición donde ocurre el mayor por primera vez
- \_\_la suma de todos los elementos

Para pensar: estos valores no podrán ser devueltos por la función de la manera usual, se los podrá tomar como parámetros y modificarlos?

- \_\_Escribir una función `fill_descending` que tome un arreglo de enteros y un entero correspondiente a su longitud y modifique el arreglo de manera tal que resulten almacenados números `fill_descending` que vayan desde la longitud del arreglo hasta 1, en orden decreciente. Escribir una función `main` desde donde se llame a `fill_descending` y donde se imprima el contenido del arreglo después de llamar a la función.

Ejemplo, si se aplica la función al arreglo `{0,0,0,0,0}` cuya longitud es 5 el arreglo modificado deberá ser `{5,4,3,2,1}`.

- \_\_Definir una simple función `print_address(int x)` cuyo único propósito sea imprimir la dirección del entero `x` que se le pasa. Crear una variable entera en `main`, imprimir su dirección ahí mismo y luego pasar la variable a `print_address`. Comparar los resultados. ¿Es el comportamiento esperado?
- \_\_Crear una función `create_integer()` que declare e inicialice un entero y devuelva la dirección de tal entero. Imprimir el valor del entero asociado a esa dirección de memoria en `main`. El resultado es código C legal? El compilador muestra alguna advertencia al querer compilar el programa? Es una operación segura?

### Arreglos multidimensionales

- \_\_Escribir un programa que declare un arreglo de enteros de 3 filas y 5 columnas, complete la primera fila con los primeros 5 múltiplos de 2, la segunda fila con los

primeros 5 múltiplos de 5 y la tercera con los primeros 5 múltiplos de 10.  
Imprimir el contenido del arreglo en forma de matriz.

- \_\_Escribir un programa que declare un arreglo de enteros de 11 filas y 10 columnas, complete las primeras 10 filas con los números del 1 al 100 y la última fila con la suma de los elementos de cada columna. Imprimir el contenido del arreglo en forma de matriz.
- \_\_Escribir un programa que declare una matriz de 5 filas y 3 columnas, con valores enteros, solicite al usuario el ingreso de sus elementos, y posteriormente genere su matriz transpuesta, un arreglo de 3 filas y 5 columnas donde se intercambiaron filas por columnas respecto a la matriz inicial. Finalmente, imprimir el contenido de ambas matrices, para lo cual definir una función que tome un arreglo, la cantidad de filas, la cantidad de columnas e imprima en pantalla su contenido. Ejemplo, las siguientes conforman un par de matrices transpuestas entre sí:

3 4 0	3 8 1 4 3
8 2 3	4 2 9 1 0
1 9 8	0 3 8 2 9
4 1 2	
3 0 9	

## Memoria dinámica y listas enlazadas

- Escribir un programa que lea un número, que indica cuántos enteros serán guardados en un arreglo, crear el arreglo para almacenar el tamaño exacto de los datos y entonces leer los enteros que serán guardados en el arreglo. Finalmente imprimir el promedio de los elementos del arreglo.
- \_\_Escribir un programa que lea enteros y los almacene en un arreglo sin conocer de antemano la cantidad de valores que se leerán. El programa deberá completar el arreglo hasta que se ingrese un número negativo, valor que no se agregará a la mencionada estructura de datos. Finalmente deberá imprimir el promedio de los elementos del arreglo.
- \_\_Repetir el ejercicio anterior pero guardando los valores leídos en una lista enlazada. Deberán implementarse las funciones:
  - \_\_insert\_element, que toma una lista y un entero y le agrega un nodo con el valor especificado al final de la lista
  - \_\_calculate\_average, que toma una lista y devuelve el promedio de los valores almacenados en la lista