



# UNIVERSIDAD NACIONAL DE INGENIERÍA

## Facultad de Ciencias

## Escuela Profesional de Ciencia de la Computación

Curso: Fundamentos de Programación CC112

Semestre 2025-I

### Laboratorio 09

## Instrucciones

Resuelve los siguientes ejercicios utilizando asignación dinámica de memoria utilizando los operadores `new` y `delete`.

1. Escribe un programa que cree dinámicamente una variable entera, le asigne el valor 10 y luego imprima su valor. Asegúrate de liberar la memoria al final.
2. Escribe un programa que pida al usuario la cantidad de números que desea ingresar, cree un arreglo dinámico para almacenarlos, los lea desde teclado y luego imprima su suma.
3. Escribe un programa que reserve memoria dinámicamente para un arreglo de  $n$  números reales, donde  $n$  es ingresado por el usuario. Llena el arreglo con valores introducidos por el usuario y luego imprime el promedio.
4. Escribe un programa que cree dos arreglos dinámicos de enteros del mismo tamaño (ingresado por el usuario), los llene con valores ingresados por teclado y calcule la suma elemento a elemento en un tercer arreglo dinámico.
5. Escribe un programa que cree dinámicamente una cadena de caracteres (estilo C) con longitud dada por el usuario. Luego, lee una cadena desde teclado y la imprime carácter por carácter.
6. Escribe un programa que permita al usuario ingresar una cantidad arbitraria de calificaciones (el usuario decide cuántas). El programa debe calcular la calificación mínima, máxima y promedio.
7. Escribe un programa que cree un arreglo dinámico de enteros y luego invierta el contenido del arreglo sin utilizar un arreglo auxiliar.
8. Escribe un programa que reserve memoria para una estructura llamada `Alumno` con campos `nombre[50]` y `edad`, inicialice sus valores y los imprima. Usa asignación dinámica para crear la estructura.
9. Escribe un programa que permita al usuario reservar dinámicamente memoria para una matriz de tamaño  $n \times m$ , llenar la matriz con valores introducidos por el usuario e imprimirla en forma de tabla.
10. Escribe un programa que cree un arreglo dinámico de cadenas (estilo C++ usando `string`) donde el usuario indique cuántas cadenas desea ingresar. Luego, el programa debe leerlas e imprimirlas en orden inverso.

11. Crea una función que reciba un arreglo dinámico de enteros y su tamaño, y devuelva un nuevo arreglo dinámico con los mismos elementos duplicados (cada elemento se repite dos veces seguidas).

**Ejemplo de entrada:** [1, 2, 3]

**Ejemplo de salida:** [1, 1, 2, 2, 3, 3]

12. Crea una estructura llamada `Libro` con los campos `titulo`, `autor` y `año`. Permite al usuario ingresar dinámicamente una cantidad de libros, almacenarlos y luego imprimir una lista de los libros publicados después del año 2000.
13. Escribe un programa que cree una matriz cuadrada dinámica de tamaño  $n \times n$ . Llena la matriz con números consecutivos empezando en 1, y luego imprime su transpuesta.
14. Implementa una función que reciba un arreglo dinámico de enteros y elimine todas las apariciones de un número dado. La función debe devolver un nuevo arreglo dinámico con los elementos restantes.
15. Escribe un programa que lea una lista de nombres (usando `string` y punteros), los almacene dinámicamente, y luego los ordene alfabéticamente usando bubble sort.
16. Escribe una función que cree dinámicamente una matriz de  $n \times m$  enteros, la llene con números aleatorios del 1 al 100, y luego cuente cuántos valores son múltiplos de 3.
17. Implementa una función que reciba dos arreglos dinámicos de enteros (de tamaños diferentes), y retorne un nuevo arreglo dinámico que contenga su concatenación.

**Ejemplo:**

A = [1, 2], B = [3, 4, 5]

Salida: [1, 2, 3, 4, 5]

18. Diseña una estructura `Producto` con los campos `nombre` y `precio`. El programa debe permitir registrar varios productos dinámicamente y luego mostrar los productos cuyo precio está entre 50 y 100.
19. Escribe una función que reciba una matriz dinámica cuadrada de enteros y determine si es simétrica respecto a la diagonal principal.
20. Implementa una agenda dinámica de contactos utilizando un arreglo dinámico de estructuras `Contacto`, donde cada contacto contiene nombre, número y correo. Permite al usuario agregar contactos, buscar uno por nombre y eliminar un contacto.
21. Diseña un programa que calcule los coeficientes del polinomio interpolador de Lagrange a partir de un conjunto dinámico de puntos ( $x$ ,  $y$ ) ingresados por el usuario.
22. Simula una red de resortes conectados en línea. Cada resorte tiene una constante elástica  $k$ . Permite al usuario ingresar el número de resortes y sus constantes, y calcula la constante equivalente del sistema:

$$\text{en serie: } \frac{1}{k_{eq}} = \sum_{i=1}^n \frac{1}{k_i} \quad \text{en paralelo: } k_{eq} = \sum_{i=1}^n k_i$$

23. Crea una calculadora de polinomios. Usa memoria dinámica para almacenar los coeficientes de dos polinomios (de grado definido por el usuario) y realiza las operaciones de suma, resta y multiplicación entre ellos.

24. Escribe un programa que implemente una pila (`stack`) usando una lista enlazada con nodos creados dinámicamente. Implementa las funciones `push`, `pop` y `top`.
25. Implementa un programa que permita crear una matriz triangular superior o inferior con asignación dinámica eficiente (usando solo memoria para los elementos necesarios). Permite realizar operaciones de suma o multiplicación escalar sobre dicha matriz.