

# EJERCICIOS PRACTICA RECU - PROGRAMACIÓN 1

## EJERCICIO 1: FUNCIÓN CON PARÁMETROS POR REFERENCIA

Escriba un programa completo que:

- Declare una función `calcularOperaciones` que reciba dos números enteros y calcule:
  - Su suma (parámetro por referencia)
  - Su producto (parámetro por referencia)
  - El mayor de los dos (parámetro por referencia)
- En el main, pida al usuario dos números
- Llame a la función y muestre los tres resultados

**Ejemplo de salida:**

Ingrese dos números: 8 12

Suma: 20

Producto: 96

Mayor: 12

---

## EJERCICIO 2: ARREGLO Y BÚSQUEDA CON PUNTEROS

Escriba un programa que:

- Declare un arreglo de 8 números enteros
- Pida al usuario que ingrese los 8 números
- Implemente una función `VOID buscarElemento` que:
  - Reciba el arreglo, su tamaño, el valor a buscar
  - Use parámetros por referencia para modificar variables:
    - Si se encontró el elemento (1=sí, 0=no)
    - La posición donde se encontró (si existe)
- Muestre el resultado de la búsqueda

**Ejemplo:**

Ingrese 8 números: 12 45 23 67 34 89 15 56

¿Qué número busca? 67

El número 67 se encontró en la posición 3

---

## EJERCICIO 3: MATRIZ DE NOTAS

Escriba un programa completo que:

- Declare una matriz de 3x4 (3 alumnos, 4 materias)
- Permita al usuario cargar todas las notas
- Implemente una función que calcule y muestre:
  - El promedio de cada alumno
  - El promedio de cada materia
  - La nota más alta de toda la matriz y en qué posición está

**Ejemplo de salida:**

=== PROMEDIOS POR ALUMNO ===

Alumno 1: 7.50

Alumno 2: 8.25

Alumno 3: 6.75

=== PROMEDIOS POR MATERIA ===

Materia 1: 7.33

Materia 2: 8.00

...

Nota más alta: 9.5 (Alumno 2, Materia 3)

---

## EJERCICIO 4: GESTIÓN DE PRODUCTOS

Escriba un programa completo que use la siguiente estructura:

```
struct Producto {  
    int codigo;  
    char nombre[30];  
    float precio;  
    int stock;  
};
```

El programa debe:

- Permitir cargar 3 productos
  - Implementar una función buscarPorCodigo que encuentre un producto por código
  - Implementar una función actualizarStock que modifique el stock de un producto
  - Implementar una función mostrarProductosBaratos que muestre productos con precio menor a un valor dado
  - En el main, usar todas las funciones para demostrar que funcionan
- 

## EJERCICIO 5: ESTADÍSTICAS DE ARREGLO

Escriba un programa que:

- Declare un arreglo de 10 números enteros
- Permita al usuario cargar los números

- Implemente una función calcularEstadisticas que use parámetros por referencia para retornar:
    - El valor mínimo y su posición
    - El valor máximo y su posición
    - El promedio de todos los números
    - Cuántos números están por encima del promedio
  - Muestre todos los resultados en el main
- 

## EJERCICIO 6: SISTEMA DE EMPLEADOS

Defina una estructura Empleado con: legajo, nombre, sueldo básico, horas extras, sueldo total.

Escriba un programa que:

- Permita cargar datos de 4 empleados (sin sueldo total)
  - Implemente una función que calcule el sueldo total:  $\text{sueldoBasico} + (\text{horasExtras} * 150)$
  - Implemente una función que encuentre al empleado con mayor sueldo
  - Implemente una función que muestre todos los empleados con sueldo superior a \$50000
  - Use todas las funciones en el main
- 

## EJERCICIO 7: MANIPULACIÓN DE MATRIZ CON FUNCIONES

Escriba un programa que:

- Declare una matriz de 4x4 números enteros
  - Permita al usuario cargar todos los valores
  - Implemente las siguientes funciones:
    - sumarDiagonalPrincipal: retorna la suma de la diagonal principal
    - contarPares: cuenta elementos pares en toda la matriz
    - encontrarMaximoPorFila: llena un arreglo con el máximo de cada fila
  - Muestre todos los resultados
- 

## EJERCICIO 8: BIBLIOTECA DE LIBROS

Cree una estructura Libro con: código, título, autor, año, disponible (1=sí, 0=no).

Escriba un programa completo que:

- Maneje un arreglo de 5 libros
- Implemente función cargarLibro para ingresar datos
- Implemente función buscarPorAutor que muestre todos los libros de un autor

- Implemente función prestarLibro que cambie disponibilidad si está disponible
  - Implemente función mostrarDisponibles que liste libros disponibles
  - Use todas las funciones en un menú simple
- 

## EJERCICIO 9: ANÁLISIS DE VENTAS MENSUAL

Escriba un programa que:

- Use una matriz de 12x4 (12 meses, 4 vendedores)
  - Permita cargar las ventas de todo el año
  - Implemente funciones que calculen:
    - Total de ventas por vendedor (todo el año)
    - Total de ventas por mes (todos los vendedores)
    - El mes con mayores ventas totales
    - El vendedor con mejores ventas anuales
  - Use parámetros por referencia donde sea necesario
  - Muestre un reporte completo
- 

## EJERCICIO 10: SISTEMA INTEGRADO DE ESTUDIANTES

Cree las estructuras:

```
struct Materia {  
    char nombre[30];  
    int creditos;  
    float nota;  
};
```

```
struct Estudiante {  
    int legajo;  
    char nombre[50];  
    struct Materia materias[5];  
    int cantidadMaterias;  
};
```

Escriba un programa completo que:

- Maneje un arreglo de 3 estudiantes
  - Implemente función cargarEstudiante completa
  - Implemente función calcularPromedioCreditos que calcule promedio ponderado por créditos
  - Implemente función buscarMejorEstudiante que encuentre el de mejor promedio
  - Implemente función materiasAprobadas que cuente materias  $\geq 6.0$  de un estudiante
  - Implemente función reporteCompleto que muestre toda la información ordenada
  - Cree un main que use todas las funciones y muestre un reporte final
-