

# Algoritmos y Estructuras de Datos II

Laboratorio 2025 - Recuperatorio 1er Parcial

## Requerimientos:

1. **Debe compilar.** Si no compila, no se aprueba el ejercicio.
2. **Debe pasar los tests.** Si no pasa los tests, no se aprueba el ejercicio.
3. **El código debe ser prolijo y comprensible, indentado y comentado.** Si no, baja nota.

## Código kickstart

Formulario de entrega: <https://forms.gle/QUchHKnUaBJr7n5W6>

## Ejercicio 1: Insertion Sort Intercalado

Archivos entregables: `intercalated_sort.c` `tests.c`

Considere el algoritmo de ordenación por inserción visto en el teórico:

### Ordenación por inserción

Todo junto

```
proc insertion_sort (in/out a: array[1..n] of T)
  for i:= 2 to n do
    insert(a,i)
  od
end proc

proc insert (in/out a: array[1..n] of T, in i: nat)
  var j: nat
  j:= i
  do j > 1 ∧ a[j] < a[j - 1] → swap(a,j-1,j)
    j:= j-1
  od
end proc
```

**Ejercicio 1.1:** Implementar en `intercalated_sort.c` el algoritmo de **ordenación por inserción** pero que ordena **solamente las posiciones pares del arreglo** (*0-based*, o sea empezando de la posición 0).

Compilar y testear con:

```
$ gcc -Wall -Wextra -std=c99 tests.c intercalated_sort.c -o tests
```

```
$ ./tests
```

**Ejercicio 1.2:** En `tests.c` se incluyen 18 casos de test. **Inventar y agregar cinco (5) nuevos casos de test** con arreglos de al menos cinco (5) elementos.

## Ejercicio 2: Más cálculos sobre datos climáticos

**Archivos entregables:** `queries.c`

Como en el laboratorio 3, en el archivo `input/weather_cordoba.in` se proveen datos climáticos históricos de Córdoba para los años comprendidos entre 1980 y 2016. Cada línea del archivo contiene números enteros con los datos de un día, con el siguiente formato:

`<año> <mes> <día> <t_media> <t_max> <t_min> <pres> <h> <prec>`

Por ejemplo la línea:

`1982 2 8 200 256 168 10148 77 0`

indica que el 8 de febrero de 1982 se dieron las siguientes mediciones:

- temperatura media `t_media`: 20.0 grados
- temperatura máxima `t_max`: 25.6 grados
- temperatura mínima `t_min`: 16.8 grados
- presión atmosférica `pres`: 1014.8 hectopascales
- humedad `h`: 77%
- precipitaciones: 0 milímetros

**Por simplicidad, se incluyen datos solamente de los días 1 a 28 de cada mes.**

**Ejercicio 2.1:** Implementar en `queries.c` la función `max_daily_rainfall()` que, dados los datos climáticos y un año calcula **para cada mes de ese año la máxima precipitación en un día**.

Compilar y testear con los comandos:

```
$ gcc -Wall -Wextra -pedantic -std=c99 weather.c weather_table.c queries.c tests.c -o tests
$ ./tests
```