



DEPARTAMENTO DE CIENCIAS DE LA COMPUTACIÓN  
ESCUELA DE INGENIERÍA  
PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE

---

IIC2343 - Arquitectura de Computadores (II/2025)

## Guía de ejercicios: RISC-V

Ayudantes: Daniela Ríos (danielaarp@uc.cl), Alberto Maturana (alberto.maturana@uc.cl), Ignacio Gajardo (gajardo.ignacio@uc.cl)

### Pregunta 1: DCCálculo - RISC-V al rescate

Te encuentras estudiando para un ramo de la Universidad, cuando en un ejercicio de ayudantía te aparece la temible función **factorial:  $x!$** , la cual te toma mucho tiempo calcular y no alcanzarás a terminar tu estudio a tiempo.

Por suerte, también te encuentras cursando Arquitectura de Computadores y, como expert@ en RISC-V, decides elaborar un código capaz de calcular esta función al instante y así agilizar tu estudio. Para esto, puedes utilizar el siguiente código base:

```
.data
x: .word 4

.text
# Completa tu código acá,
```

**IMPORTANTE:** No es necesario que respetes la convención de llamadas en este ejercicio.



## Pregunta 2: DCCálculo - Resumen perdido

Llegó el día de dar la prueba para la que tanto has estudiado y, al mirar Canvas, ¡te enteras que podrás llevar una hoja de resumen! Sin embargo, entre todo el caos del estudio no logras encontrar la hoja en tu cajón de apuntes. Por suerte, tienes todas tus hojas marcadas con un identificador, por lo que decides utilizar tus habilidades de RISC-V y con ellas crear un programa que te ayude a encontrar el resumen.

Para esto, deberás elaborar un programa que realice la **búsqueda binaria** de un elemento **x** sobre un arreglo ordenado **arr** y guardar su índice en **element\_index**. Si el elemento **x** no se encuentra en **arr**, entonces se debe mantener por defecto el valor **element\_index = -1**.

La búsqueda binaria se realiza a través del siguiente procedimiento:

- Se revisa el elemento central del arreglo `arr[left_bound : right_bound]`. Llamaremos `c` al índice del elemento central y se computa así: `c = (left_bound + right_bound) // 2`.
- Si el valor del elemento central es igual al valor buscado, *i.e.* `arr[c] == x`, se retorna su posición como el resultado de la búsqueda.
- Si su valor no es igual al buscado:
  - Si el elemento central es menor al elemento buscado, se realiza la búsqueda nuevamente con `left_bound` igual a `c + 1`.
  - Si el elemento central es mayor al elemento buscado, se realiza la búsqueda nuevamente con `right_bound` igual a `c - 1`.
- La búsqueda termina cuando `left_bound` es mayor a `right_bound`.

Puedes utilizar el siguiente fragmento de código como base:

```
.data
arr:      .word -1000, -255, -7, -1, 0, 10, 11, 27, 255, 1000, 10000 # Arreglo
left_bound: .word 0 # Límite izquierdo
right_bound: .word 10 # Límite derecho
x:          .word -7 # Elemento a buscar
element_index: .word -1 # Índice del elemento

.text
# Tu código aquí
```

**IMPORTANTE:** No es necesario que respetes la convención de llamadas en este ejercicio.

## 1. *Feedback* cápsulas

Escanee el QR para entregar *feedback* sobre las cápsulas.

