



TÉCNICO
LISBOA

Homework 1

Computer Architecture and Organization

Arquitetura e Organização de Computadores (AOC)

2025/2026

Overview

This assignment aims to strengthen your understanding of the **RISC-V Instruction Set Architecture (ISA)** and how to write assembly programs using the [Ripes](https://github.com/mortbopet/Ripes)¹ **simulator**. In this assignment, you will:

1. Develop practical skills in **assembly programming** for RISC-V processors
2. Explore the **interaction with the console**, i.e., how to print **ASCII strings** using **system calls** (`ecall`)
3. Understand how **data** and **instructions** are organized and represented in **memory**.

Deadlines and Submission

This homework is practice-oriented and tutorial-based, intended to be developed at home.

The content provided in this assignment is accompanied by the *Homework 1 Response Sheet*, which includes the tutorial, specific questions to answer, and tasks to complete. In addition to analyzing a provided starter code, you will also need to develop simple RISC-V assembly codes.

You should fill in your answers to the questions in the respective fields of the response sheet (available on the course webpage, section: Labs). The completed response sheet in PDF format, along with the developed (final version of your) RISC-V code in `.s` format, should be placed in a single `.zip` file and submitted on Fenix by **Wednesday, November 26, 2025, at 23:59**.

Important: You must also append your final code at the end of your response sheet (PDF file). Do not forget to **comment** your source code (both in the `.s` source file and in the annex).

Your responses and codes will be evaluated for correctness and optimization efforts. Your codes must run out-of-the-box in the Ripes simulator and produce the correct result (with no undesirable side effects). Otherwise, your solution will not be graded, regardless of its efficiency!

You should submit **only one** final version of your code (multiple versions will not be graded!). Avoid using questionable tricks or exploiting specific functionalities of the Ripes simulator that were not covered in class—if in doubt, ask us! Always aim to develop general and reusable codes (unless otherwise stated).

Helpful hints: You should aim to create code that:

- Takes the minimum number of clock cycles to execute.
- Uses the minimum number of instructions.
- Minimizes register usage.
- Reduces access to main memory.

Of course, you should not overcomplicate things!

¹ <https://github.com/mortbopet/Ripes>

Program to execute and develop

In this assignment, we will analyze how ASCII characters can be stored and processed, as well as how data and instructions are organized. You will also have the opportunity to apply your knowledge of the RISC-V ISA and basic assembly programming skills.

You are given a string of six (6) input ASCII characters stored in the Keys array. Your task is to analyze the code and combine these characters to produce the messages **"HI!OLA!"** (uppercase, no spaces) and **"hi!ola!"** (lowercase, no spaces) in the Output array. The starter code is provided in the homework_1.s file, and detailed instructions on the work to be performed are outlined in the *Homework 1 Response Sheet*.

```

/*****
/***** RISC-V CODE *****/
/*****/

### Data segment
.data
Output: .zero 8
Keys: .string "!AHILO"

### Program segment
.text

### TODO ###
# here you should provide
# your code for QUESTION 6
#####
##### PRINT ECALLS #####
##### !!! DON'T TOUCH !!! #####
li a0, 0xA # new line char ecall
li a7, 11 # print char
ecall
la a0, Output # string to print
li a7, 4 # print string ecall
ecall
#####

### TODO ###
# here you should provide
# your code for QUESTION 7
#####
##### PRINT ECALLS #####
##### !!! DON'T TOUCH !!! #####
li a0, 0xA # new line char ecall
li a7, 11 # print char
ecall
la a0, Output # string to print
li a7, 4 # print string ecall
ecall
#####

# INSTRUCTIONS TO CONSIDER
# FOR QUESTION 4
auipc x5, 0
jalr x0, x5, 0 # the same as jalr x0, 0(x5)

```

What do I need to do?

As previously mentioned, you should begin by completing the tutorial and answering the questions provided in the *Homework_1_Response_Sheet* (available on the course webpage, section: Labs). Once you are satisfied with your work, submit it on Fenix.

Good luck!