



INTRODUCTION TO COMPUTERS II

LAB 5: PROJECT

1. Implement your own program, using the **pr5** project in the Workspace. The program must meet the following requirements in **Assembly language**:
 - It must have a **main** function, which would be the entry point to the program.
 - Input and output data must be declared in assembly, using the appropriate sections.
 - It must use at least one conditional structure in assembly (if).
 - It must use at least one loop structure in assembly (for/while) to traverse an array and perform some calculations on its elements.
 - Apart from **main**, there must be two additional assembly functions: a leaf function and a non-leaf function that calls the first one.
 - At some point, the program must pass scalar values and arrays as arguments to a function.

It is mandatory to add enough **comments** to the program in order to facilitate readability. Also, you must provide a **clear description** of what the program does and how it does it.

The program should have an application to actual problems, as e.g. sorting, searching, vector calculations, matrix calculations, access to structures or other algorithms working on vectors. **If in doubt, please talk to your professor.**

The code must be developed specifically for this project, and therefore it cannot be reused from the previous lab projects. However, it can be based on already existing C code, which will have to be translated into assembly language (although it should not have system calls as **printf / cin / cout** or external libraries).

2. Optionally, C code can be added around the developed assembly program. A valid option is to choose an already implemented C algorithm and translate part of it into assembly. Remember that the entry point must be programmed in assembly, as well as the leaf and non-leaf functions.
3. **Evaluation.** The knowledge on the developed program and the ability to show results in registers and memory, using the Eclipse environment, will be evaluated. Different test cases will be required in order to demonstrate the correctness of the program.
4. As an example of what could be done in this project, a sort algorithm of the **TheAlgorithms** suite (<https://github.com/TheAlgorithms/C/tree/master/sorting>) could be chosen, creating an entry point in assembly and translating a couple of functions (leaf and non-leaf). The rest of the algorithm can be in C.