# Exercise 10


- Author: Mieszko Wawrzyniak 243563

- Date: 24 May 2018
- Version a
- Version b - with swap

## Task

The purpose of this exercise is to familiarize with the implementation of procedures (functions) in the processor MIPS assembly language.

### The task to perform
Write a program that sorts the given sequence of integers using bubble-sort algorithm.

1. Structure of the program should be as follows: "The master" is responsible for communication with the user - input and output of the program. Sorting algorithm should be implemented as a procedure called from the main module.
2. Modify the program written in paragraph (a) in such a way that swap of two elements is carried out by a procedure called by the sorting procedure.
3. What are the differences in the versions (a) and (b), of course, except for an additional procedure ?

## Program description

Program works performing the following steps:

- Reads array length.
- Allocates memory for the array on the heap.
- Prints provided array.
- Sorts the array.
- Prints sorted array.

In order to preform these tasks program implements following procedures:

- `read` which reads array elements from standard input.
- `sort` which sorts an array of integers using bubble sort algorithm.
- `print` which prints array elements.
- `swap` which swaps to given integers in RAM. This procedure is only implemented in the second program.

  In the case of the second version with additional `swap` procedure procedure `sort` stores the value of the `$ra` on the stack due to nested procedure calls. This operation is required because `jal swap` overrides `$ra` register.

# Conclusions

- In the case of nested procedure calls storing the value of `$ra` is required in order to keep program running correctly.
- The best place for storing `$ra` value is a stack.
- `$ra` value should be stored right before `jal` operation and restored right behind it.