

# Assignment 1 - Arrays, Loops and Functions

## Academic Honesty

The assignment is individual work. We use MOSS to detect software plagiarism.

Students are allowed to consult books and online resources but must acknowledge the resources used in the report.

## Problem 1: Unsorted arrays

Given an array of integers that is unsorted, implement the following functions:

- **myAdd ( )**: add an integer  $d$  to the array; return 0 if the operation is successful; return a negative number otherwise.
- **myRemove ( )**: remove integer  $d$  from the array; return 0 if the operation is successful; return a negative number otherwise (e.g.,  $d$  is not found in the array).
- **search ( )**: given an integer  $d$ , if  $d$  is found in the array, return the index of the cell containing  $d$ . Return a negative number otherwise (e.g.,  $d$  is not found in the array).

Notes:

- The functions (algorithms) must be the most efficient in terms of running time and memory usage.
- Submit file [arrayUnsorted.c](#). Complete the header with your student and contact information. Include sufficient comments in your code to facilitate code inspection.
- Submit a report in PDF with following information: references (sources); error conditions and actions taken; brief algorithm; running time of the function (algorithm) and a brief explanation. See the template [a1report.docx](#) for an example.
- See file [a1output.txt](#) for sample output.
- Your code will be graded automatically by a script, so make sure to strictly follow the specifications.
- **Do not use any output statements (for example, printf) in your code, or they will mess up the grading scripts.**
- Use the `main( )` function provided to test your code. Understanding the code in the `main( )` function is part of the assignment. Do not change the code in the `main( )` function in the final submission, or your program will mess up the grading script.

## Problem 2: Sorted arrays

This problem is the same as the above problem, except that the array of interreges is **sorted** and must always be maintained in **non-decreasing order** at any point in time. *Note*: Do not use any sorting algorithm in the program.

Submit file [arraySorted.c](#) and a report in file [a1report.docx](#) as described above.