

# CMP\_SC 4050/7050 Homework 2

## Due: 11:59 pm, 9/21/2022

The first homework of the semester is a programming assignment. The homework is out of 30 points. The first part of the homework specifies the exact problem that your submission should solve and the second part describes the constraints that you **must** follow (no exceptions).

### Specification:

For this programming assignment, one compressed file containing the directory of all of the assignment files should be submitted to CANVAS prior to the due date. Your main-level should be named as pawprint1\_pawprint2\_hw2.c where you should replace pawprint1 and pawprint2 by the pawprints of the team members. For example, if students with pawprints Ryan and Mathilde are group members, the file should be names as Ryan\_Mathilde\_hw2.c or Mathilde\_Ryan\_hw2.c. In case, you are working along, you should use your pawprint twice in the name. Your program should compile on a UNIX machine.

The assignment should:

- Take one filenames as input in the command line. The file is going to be the input to the program.
- Each input file represents a list of non-negative integers. Each line of an input file consists of a non-negative integer followed by a newline character. Each integer is a string of digits. One sample input file can be found on CANVAS.
- Your program should implement the linear-time selection algorithm (the median of medians algorithm) discussed in class.
- Your program should output on the standard output the “x”-th smallest number in the list of numbers in the input file starting from line number 2 followed by a newline character, where “x” is the number on the first line. For example, if the input file contains the integers 3, 2, 10, 1, 1, 2, 3, 2 then your program should output number 2\n. If there are less numbers than “x” then the program should output -1\n.
- It is important that your output only contains the “x”-th smallest number followed by the newline character. For example, if the input file contains the integers 3, 2, 10, 1, 1, 2, 3, 2 then your program should only output number 2\n. It is absolutely essential that you do not put any whitespaces or any other character in that line.

## Constraints:

The following is a list of constraints for this assignment; failure to adhere to any of these constraints will result in a loss of points or even a zero.

- The assignment should be completed in C.
- Built-in data structures and external libraries other than standard input/output functions may not be used for this assignment. If the program requires a stack, linked list, or any other structure, it is up to you to provide it. If you have any doubt on whether or not any technique you wish to use is acceptable, do not hesitate to ask.
- A moderate amount of error checking and resource management is required. Even if you do some error recovery, you must report errors in the input files. Your application should ensure that each line from the input file is properly formatted, that the file is successfully opened, the file is successfully closed upon reading of the file and that all allocated space is de-allocated at the exit.
- For reporting input format error, please use the enum type provided in `input_error.h`. Please do not alter the file `input_error.h`. You need to include `input_error.h` in your main program. The file `input_error.h` contains documentation of how to use `input_error.h`. Please do not report anything else on the standard output.
- A moderate amount of formatting and documentation is required. Comments should be descriptive and used to illustrate the purpose and inner workings of an algorithm or function; they should not be used to annotate each line or self-evident logic.
- The assignment must be submitted via CANVAS. Please compress the folder before submitting using zip.
- The input size is unknown, and the input range is 0 to `UINT_MAX`. This means that your program should not make any assumptions about the range or size of the input.
- You must write your own test-cases. We will check your program against our own test-cases. To receive any credit, your program must pass at least one of the tests. For full credit, you would have to pass all of our test-cases.

## Timing Constraints

A good algorithm for this homework should run in  $O(n)$  where  $n$  is the number of integers in the input file. While we shall not be grading the exact time complexity of your solution, we shall be timing your solution and grading efficiency. As a guideline, on an input file consisting of million integers, your program should run under a minute of system time to receive full credit for efficiency.

## Grading:

There are 30 points possible for this assignment. The grade breakdown is as follows:

- 4 points for error checking and resource management.
- 4 points for general programming style and adherence to the constraints.
- 18 points for correct outputs.
- 4 points for efficiency.

If the program fails to compile or crashes due to a runtime exception, a grade of zero will be assigned.