University of Cyprus
Department of Electrical and Computer Engineering

ECE 316 - Operating Systems and Networks Laboratory

Assignment 1

General Instructions:

For each Assignment, a report (.pdf) and the source-code (.c or .cpp) of the solution should be submitted through Microsoft Teams "ECE316 - 2022 - Operating Systems and Networks Laboratory" not later than the due date of the Assignment. The report should start with a cover page that clearly contains the assignment number and the Team number and the names of the members. In your report, include only the pseudocode, not the actual code, with any comments and description you may want to add, as well as a typical scenario that you used to test your programs. Please note that the report should be as concise as possible. Caution: You are not allowed to upload executables (.exe) or zipped files (.zip or .rar)!

Deliverables:

- Each Member of the Team should upload a different report that illustrates the outcomes of the exercises as resulting from executing the procedures on a personal computer.

If input test files are given, you are not allowed to make any changes to the provided input files.

Report File Naming Format: "Team#_Assignment#.pdf"

Description:

This assignment is a revision for reading and parsing a file, which we are going to use at a later stage. You should write a program in C/C++ that will read from a file and parse it based on a delimiter. It should then execute calculations based on the given inputs. Notice: You can find an example of how to parse a file in https://tinyurl.com/file-parse-c.

Exercises 1 **[35%]**:

The file commands will be in the format (see **"Input_TestFile_Exe1.txt"**):

```
------------------------------
Add#3,4,100,223
Subtract#35,60,1
Multiply#3,15
Subtract#10000,500,29
------------------------------
```

For this case the delimiter is "#" for the operators and "," for the values. Note that there can be any number of arguments per line. Your program should print every line that it reads, do the calculations and then proceed to the next line. An example of the expected output for the previous commands appears below:

Exercises 2 **[65%]**:

The file commands will be in the format (see **"Input_TestFile_Exe2.txt"**):

```
------------------------------
Remove,100
Add,100
Add,200
Print
Size
Add,400
Add,300
Print
Remove,400
Remove,100
Print
Size
Remove,200
Print
Size
------------------------------
```

For this case the delimiter is "," and your program should read each line from the file and parse it. It should then execute the following commands based on the first word of each line:

Add: add integer/s, may have any number of arguments which must be added in a FIFO manner to the queue (one after the other)

Remove: remove the integer based on the FIFO order (integer at the head). If no argument is given remove the integer that is located at the head without checking. Remove works unless the queue is empty. In that case it should print an appropriate message.

Print: print the contents of the queue

Size: print the size of the queue (e.g. "The size of the queue is x")

An example of the expected output for the previous commands appears below:

```
Element 100 could not be removed because the queue is empty
The contents of the queue are the following : 100,200
The size of the queue is 2
The contents of the queue are the following : 100,200,400,300
Element 400 could not be removed
The contents of the queue are the following : 200,400,300
The size of the queue is 3
The contents of the queue are the following : 400,300
The size of the queue is 2
```