



ECE 316 - Operating Systems and Networking Laboratory

Practical Assignment 1 (Due: 22/01/2020)

Deliverables: Your report and code should be sent via email to lab teaching assistants (aanast01@ucy.ac.cy or ahadji29@ucy.ac.cy) prior the assignment examination and must include the usual cover page. 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 use to test your programs. Email subject line should only consist of "ECE316_2020". Naming format for the zip file: lastName.zip (or lastName_lastName.zip if a group). **Caution:** Remove the executables (.exe) from the files before you send them!

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>.

1. [35%] The file commands will be in the format:

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

For this case the delimiter is ",". 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:

```
The equation is:  
(+3+4+100+223-35-60-1)*3*15(-10000-500-29)  
The result is: 1
```

2. [65%] The file commands will be in the format:

```
-----  
Remove,100  
Add,100  
Add,200,300  
Print  
Size  
Add,400  
Add,500  
Print  
Remove,400  
Remove  
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.  
100 is added to the queue  
200 is added to the queue  
300 is added to the queue  
100 ← 200 ← 300  
The size of the queue is 3  
400 is added to the queue  
500 is added to the queue  
100 ← 200 ← 300 ← 400 ← 500  
Element 400 could not be removed.  
100 has been removed  
200 ← 300 ← 400 ← 500  
The size of the queue is 4  
200 has been removed  
300 ← 400 ← 500  
The size of the queue is 3
```

