School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg
ELEN2020A - Software Development I

Lab 3 - Arrays and functions

Lab 3 consists of 3 tasks. Each week, one task must be completed and submitted. Check the due date of each task on Ulwazi. Note that the following test and submission instructions apply for all tasks.

Testing your code

You should refer to the provided set of inputs and outputs listed in the lab brief to determine if your code is running correctly. Given the set of inputs your code should be able to reproduce the output provided. Make sure to test it with additional values not in the lab brief to ensure it is working correctly.

Submission instructions

Your source code (.cpp) must be submitted to the *Lab 3 - task X* link under *Assignments* on Ulwazi (where `X` is the task number). **Only submit your source code.**

It is important that you follow **good programming practices** in your code. This includes

- indent code correctly (use the AStyle formatting tool in Code::Blocks),
- add necessary comments and choose meaningful names of variables in your code,
- add the date and author at the top of your source code. Add the following comment at the start of each source code file.

```
// Student number:
// Date:
```

Task 1 (week 1) - Sort a list using a function

Write a program to sort a list of numbers of unknown length. Your program must

- read in a list (of unknown size) of numbers (think about types) from an input file named **`input.txt`**,
- use a function, written by you, to sort the list in ascending order and
- write the sorted list to an output file named **`output.txt`**.

First draw a flow chart and then attempt the code.

Example 1

Example input and output files can be seen in Listings 1 and 2.

Listing 1: Example of **`input.txt`**

```
-2
3
8.25
1
9.34
100
98
99.999
```

Listing 2: Example of **`output.txt`**

```
-2
1
3
8.25
9.34
98
99.999
100
```

Example 2

Example of additional input and output files can be seen in Listings 3 and 4.

Listing 3: Example of **`input.txt`**

```
10
9
7
8
4
5
2
1
```

Listing 4: Example of **`output.txt`**

```
1
2
4
5
7
8
9
10
```

Task 2 (week 2) - Statistics

Write a program that computes the standard deviation of a given list. The standard deviation is given by

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2} \tag{1}$$

where N is the number of samples in the list, x_i is the ith sample and μ is the mean (average) of the list given by

$$\mu = \frac{1}{N}\sum_{i=1}^{N} x_i. \tag{2}$$

Your program must

- read in a list (of unknown length) of numbers from a file named **`input.txt`**,
- using a function written by you, calculate the standard deviation for the list and
- write the standard deviation to 4 decimal places to a file named **`output.txt`**.

First draw a flow chart and then attempt the code.

Example 1

Example input and output files can be seen in Listings 5 and 6.

Listing 5: Example of **`input.txt`**

```
5.5
6
7
9.9
3.3
4
6.7
```

Listing 6: Example of **`output.txt`**

```
2.0077
```

Example 2

Example of additional input and output files can be seen in Listings 7 and 8.

Listing 7: Example of **`input.txt`**

```
202.4
198
219.2
200
199.9
400
212.212
```

Listing 8: Example of **`output.txt`**

```
68.5122
```

Task 3 (week 3) - Frequency of elements

Write a program that determines the frequency of each char in an input file. Your program should

- read in an unknown number of **`char`**s from an input file named **`input.txt`**,
- using a function written by you, determine the frequency of occurrence of each **`char`** in the list
- write each **`char`** followed by the frequency to the output file **`output.txt`**.
- The **`chars`** should appear in the output file in the same order as they first appeared in the input file.

First draw a flow chart and then attempt the code.

Example 1

Example input and output files can be seen in Listings 9 and 10.

Listing 9: Example of **`input.txt`**

```
a
a
f
e
v
x
a
v
f
f
f
g
```

Listing 10: Example of **`output.txt`**

```
a 3
f 4
e 1
v 2
x 1
g 1
```

Example 2

Example of additional input and output files can be seen in Listings 11 and 12.

Listing 11: Example of **`input.txt`**

```
a
b
a
b
a
b
b
```

Listing 12: Example of **`output.txt`**

```
a 3
b 4
```