TIC1002: Introduction to Computing and Programming II
Semester II, 2018/2019

Self-Check Exercise

This Exercise mimic a typical 2-hour practical exam with content only from TIC1001. Try to time yourself on the time spent to gauge your proficiency.

Task 1. Lies, Wicked Lies and ……

Your boss just received a long list of monthly sales data. As her favorite "problem solver", you are tasked to compute the **average** and **standard deviation** for the data.

Implement a function where

```
void compute_stats( int array[], int size,
                    double* average, double* stddev);
```

array[] is an array (duh!) with **size** number of integer data

average and **stddev** are two pass-by-address variables for storing the **average** and **standard deviation** of the data in array[]

Average (also known as **mean,** $\bar{x}$) is computed as:

$$\bar{x} = \frac{1}{n}\left(\sum_{i=1}^{n} x_i\right) = \frac{x_1 + x_2 + \cdots + x_n}{n}$$

where $\mathbf{X_i}$ is the value at **array[i]**, and **n** is the number of data.

Standard deviation is computed as:

$$s = \sqrt{\frac{\sum_{i=1}^{N} (x_i - \overline{x})^2}{N-1}}$$

where $\bar{x}$ is the average of the data set.

Restriction
You are allowed to use only sqrt() (square root) function in your implementation.

Sample Result

Data	Average	Standard Deviation
{1, 2, 3, 4, 5, 6, 7, 8}	4.5	2.4495

Credit: Both formula images were taken from Wikipedia.

Task 2. This is almost… unique

The **United Earth Starfleet (UES)** found a new sentient species Rrrrr at the outer reach of Solar System. The Rrrrr's speech is very hard to understand as they tend to repeat some characters many ***many*** times. The commander of UES asked you to *quietly* write a program to filter out the repeated characters from Rrrrr's speech. With your help, hopefully UES will appear to be non-speciesist in their dealing with the Rrrrr…..

Implement a function where

`void n_unique( const string& input, string result, int nCopy );`
`input` contains a string of characters to be processed. **`result`** contains a result of the processed string . **`nCopy`** is the maximum repetitions allowed.
Assumption / Restrictions: • **`input`** contains only small letters and non-alphabets characters, i.e. no capital letters. • The filtering only applies to **alphabets**. All non-alphabets are retained. • **`result`** must contain the allowed alphabets in the same order as the **`input`**.

Sample Results

original	nCopy	result
"abcdef!!abc, cba defa bcaba."	1	"abcdef!!, ."
"abcdef!!abc, cba defa bcaba."	2	"abcdef!!abc, def ."
"abcdef!!abc, cba defa bcaba."	3	"abcdef!!abc, cba def ."
"abcdef!!abc, cba defa bcaba."	4	"abcdef!!abc, cba defa bc."

Brief Explanation

When nCopy == 1, each alphabet can appear **only once**. Hence, you can see that the result filtered out all repetitions of "abcedef" after the first occurrence.

When nCopy == 3, each alphabet can appear **at most three times.** You can see that the result contains most of the letters from the original, up until the "……cba def", all subsequent letters were dropped as they exceed the 3 repetitions constraint.