**Introduction to Programming – Fall 2017**

**Assignment 1**

**Notes:**

1. You may discuss concepts and ideas for solving the problems with others, but everyone must submit their own programs (no group submissions). Feel free to discuss with Mark or me in case you get stuck, have doubts, need clarification or some high-level guidance to refine your approach.
2. In order for it to be considered for grading, the program you submit **\*\*must\*\*** run and produce output. For example, we should be able to go to the shell/Terminal or PyCharm and execute your program to see the results. Partial grading (in case of partially correct answers) will only be done if this condition is met.
3. E-mail an attached copy of your scripts or a link to a github repository containing your scripts.
4. **Due date:** 02 October 2017, by 11:59 pm.

**Question 1:**

You are required to create a program which accepts DNA sequences as input and stores them in a list. These sequences consist of only ['A/a', 'C/c', 'G/g', 'T/t'] characters, but there is no guarantee that the user will not make mistakes. Once a sequence has been entered your program should first do a quality control step to ensure that the sequence is valid (only contains [A/a, C/c, G/g, T/t] characters). If the sequence is not valid display an error message. If the sequence is valid reverse complement it (Reverse the sequence and swap ‘A/a’ characters with ‘T/t’, ‘G/g’ characters with ‘C/c’ and vice versa) If the resulting sequence is already in the list of sequences alert the user that this is the case otherwise store the sequence in the list. The program should start by prompting the user for the number (**n**) of sequences which will be entered. The user will then enter **n** sequences one at a time. Once all the sequences have been entered and evaluated the valid reverse complemented sequences should be printed.

Optional: Print some statistics regarding the sequences.

eg:

* frequency of each of the valid characters
* number of sequences and the average length of each sequence
* come up with something you think would be helpful for the user

**pseudocode:**

get number (**n**) of sequences to be entered by user

initialize variable to hold sequences

loop **n** times:

* get sequence from user
* check whether the sequence is valid
  + alert user if invalid
* reverse complement the sequence
* check whether sequence is a duplicate
  + alert user if this is the case
* add sequence to list if it is valid and unique

print list of sequences

**Example:**

**Use case 1** - Incorrect character is input in the sequence

user input: ACGTB

**output**: ‘one of the characters is invalid’ or (‘B is an invalid character’ - this is more complicated to do)

**Use case 2** - valid sequence is input and the reverse complement is stored

user input: AGCCT

**output**: ‘AGGCT’ has been added to the list

**Use case 3** - duplicate sequence is input

user input: AAGCT

**output**: ‘AGCTT’ has been added to the list

user input: AAGCT

**output**: ‘AGCTT’ is already in the list

**Use case 4** - another valid sequence is input and reverse complement is stored

user input: AGCT

**output**: ‘AGCT’ has been added to the list

**Output example after all use cases:**

**output**: Sequences:

AGGCT, AGCTT, AGCT

--\* Statistics \*-- (This is optional)

frequencies:

A/a : 3/14 characters

T/t : 4/14 characters

G/g : 4/14 characters

C/c : 3/14 characters

number of sequences:

3

Average sequence length:

4.667 characters

**Question 2:**

In this problem, you’re provided a **list of** protein **dictionaries**, with each dictionary storing a protein sequence and its charge state (integer). Your goal is to calculate the ratio of mass to charge (m/z) of all the protein sequences in the list and print the results. A protein sequence is represented as a string of characters. Each character corresponds to an amino-acid (AA) molecule with a specific mass. There are 20 possible characters, one for each naturally occurring AA, and anything outside of this 20-letter alphabet is considered invalid.

The individual AA characters, their masses, and the calculation for m/z of a protein sequence, are provided as comments in 'a1\_q2.py' file. You have to implement the process of calculation in code.

Your program must do the following:

1. Represent masses in an appropriate data structure.
2. For each sequence provided in the ‘a1\_q2.py’ file:
   1. Check for validity of the sequence
   2. If valid, compute m/z, then print the sequence and its m/z. Otherwise, print the sequence and an error message (like “Invalid”).

Also, AA sequences may be represented as upper or lower case. Your program should be able to handle both.