University of the Philippines Los Baños
College of Arts and Sciences
Institute of Computer Science

CMSC 128 Introduction to Software Engineering
2nd Semester AY 2015 – 2016
Section AB

Assign 002: Programming a Simple Bioinformatics Library

By: RNC Recario

Using the assigned programming language for your section, create a library that will implement the provided functionalities as shown below.

Deadline: Mar 12, 2016 12nn Philippine Standard Time (PST)

Submission mode: An assignment link will be provided on Moodle. Put there the link of your Github repo containing your assignment. The name of the repository must be cmsc128-ay2015-16-assign002-<pl code>. For example: cmsc128-ay2015-16-assign002-js.

On Academic honesty: All students are expected to be academically honest on the submission of their assignments. Students who plagiarize and commit academic dishonesty will be given a grade of zero (0) in this activity. If the lecturer and the lab instructors see that the degree of dishonesty is overwhelming or wide scale in nature (i.e., organized and involves two or more students), the case can be elevated at the institute, college or university level. Students must at all times ensure that their code will not be copied by other students.

On working in groups and exchange of ideas: We recognize the fact that students need to exchange ideas in order also to practice your other critical faculties. However, directly copying someone's output is strictly prohibited.

On student safety: As what has been always said in this course, value your safety first. We know the importance of working extensively on academic outputs beyond prescribed hours since this is what happens in the "real" world. However, ALWAYS consider your safety a top priority. The assignment is given a head of time with sufficient time for you to work on it. Avoid going out of your dormitories, apartments and homes during the night if you do not have computers or internet connection.

On Programming Language: The following programming language will be used for each section:

Section	Programming Language	pl code
AB1L	Python	py
AB2L	(Native) Javascript	js
AB3L	C	c
AB4L	Python	py
AB5L	(Native) Javascript	js
AB6L	C	c
AB7L	Python	py

No student will be allowed to provide an answer written in other programming language except of those that has been prescribed.

Problem Proper:
On your assigned programming language, create a library (containing only the functions as described) to perform the following activities:

01 Function: int getHammingDistance(string str1, string str2)
What it does: Given two strings str1 and str2 of same length (length must never be 0 or negative!), the Hamming Distance of those two strings are the number of inversions per character need to transform str1 to str2 or vise-versa. Simply put, the Hamming Distance of two strings is the number of characters that differ in ith position from position 1 to strlen(str1).

Example:
getHammingDistance("AACCTT","GGCCTT") returns 2
getHammingDIstance("TCGGA","AAAAG") returns 5
getHammingDistance("A","AG") returns "Error! Strings are not equal!"

02 Function: int countSubstrPattern(string original, string pattern)
What it does: Given a string original and pattern, we will count the number of occurrence of pattern in original.

Example:
countSubstrPattern("AATATATAGG","GG") returns 1
countSubstrPattern("AATATATAGG","ATA") returns 3
countSubstrPattern("AATATATAGG","ACTGACTGACTG") returns 0

03 Function: bool isValidString(string str, string alphabet)
What it does: Given an alphabet string where all letters are assumed to be unique, this function returns true if the string str is a valid string based on the letters of alphabet.

Example:
isValidString("AAGGCTATGC","ACGT") returns true
isValidString("AAGGCTATGa","ACGT") returns false
isValidString("ACGT","ACGT") returns true
isValidString("ACGT101_","ACGT") returns false
isValidString("091212345","0123456789") returns true

04 Function: int getSkew(string str, int n)
What it does: Given a genome str of some length q (where q>0), it returns the number of Gs minus the number of Cs in the first n nucleotides (q>=n). The value can be zero, negative or positive. The first position is one (1) not zero(0) as we typically associate with string implementations.

Example:
getSkew("GGCCAC", 1) returns 1
getSkew("GGCCAC", 2) returns 2
getSkew("GGCCAC", 3) returns 1
getSkew("GGCCAC", 4) returns 0
getSkew("GGCCAC", 5) returns 0

05 Function: int getMaxSkewN(string str, int n)
What it does: Given a genome str of some length q (where q>0), it returns the maximum value of the number of Gs minus the number of Cs in the first n nucleotides (q>=n). The value can be zero, negative or positive. The first position is one (1) not zero(0) as we typically associate with string implementations.

Example:
getMaxSkewN("GGCCAC", 1) returns 1
getMaxSkewN("GGCCAC", 2) returns 2
getMaxSkewN("GGCCAC", 3) returns 2
getMaxSkewN("GGCCAC", 4) returns 2
getMaxSkewN("GGCCAC", 5) returns 2

06 Function: int getMinSkewN(string str, int n)
What it does: Given a genome str of some length q (where q>0), it returns the minimum value of the number of Gs minus the number of Cs in the first n nucleotides (q>=n). The value can be zero, negative or positive. The first position is one (1) not zero(0) as we typically associate with string implementations.

Example:
getMinSkewN("GGCCAC", 1) returns 1
getMinSkewN("GGCCAC", 2) returns 1
getMinSkewN("GGCCAC", 3) returns 1
getMinSkewN("GGCCAC", 4) returns 0
getMinSkewN("GGCCAC", 5) returns 0

This may be trivial but include a documentation in your code.
Also, it would be best to regularly push your work in your repository. Commit histories will also be reviewed.