CS250 Assignment 2
DNA p-distance Matrix using Classes

Date assigned: Monday, February 11, 2019
Date due: Wednesday, February 20, 2019
Points: 35

Goals for this assignment:

1. Write an object-oriented program using a class.
2. Practice modular programming by using well-defined functions.

Build a class to represent a DNA strand and the operations that can be performed on that strand. A strand should be represented in the class as a **string.**

The operations you will need to support on the strand include:

- Produce a reverse strand: create a new strand that is the reverse of the original strand.
- Produce a complement of a strand: create a new strand that is the complement of the original strand.
- Read a strand from a stream.
- Display a strand to a stream
- Calculate the P-stat of two strands*.

You may add other operations as necessary.

* The p-distance between two DNA strings (s1, s2) is the proportion of corresponding bases that differ between the two strings.

For example, if:

- S1 is: AAAACCCGGT
- S2 is: AAACCCGGGT

Then the p-distance, denoted p(s1,s2) is 2/10 or 0.2. The differing bases are the 4th and 7th ones.

Driver

The driver for this project must:

1. Read the first two DNA strands from a file, DNA.txt. The file may contain more than two strands but you will only read the first two. This file will have one DNA strand per line. All strands in this file will be the same length.
2. Read one DNA strand from the keyboard. This strand must be the same length as the strands in the file.
3. Display all three strands to the screen.
4. Display the reverse of all three strands to the screen.
5. Display the reverse of the reverse of all three strands to the screen (this should produce the original strands).
6. Display the reverse complement of all three strands to the screen.
7. Calculate the P-Stat between the keyboard strand and each file strand. Display the keyboard strand and the file strand most similar to the keyboard strand as well as the P-stat for these two strands. If both file strand produce the same P-Stat, display the first strand in the file.

Notes on design:

- You are to use object-oriented design using one class.
- For the class you will need a .h and a .cpp. You will also need a driver (named 02_DNAClasses.cpp) containing the main function. Your project must have at least 3 files in total.

Error Handling:

- If the file is invalid in anyway, print an error message and terminate. For example:
  - The file contains only zero or one strand
  - The file contains strands of different lengths
  - The file contains an invalid strand (a base other than A, T, C, or G is found).
- If the user provides a DNA strand of the wrong length at the keyboard or provides a strand with an invalid base, continue asking for the DNA strand until a correct, usable strand is provided.

Sample File:

```
AAA
CAT
```

Sample Output

```
***********************
* PACIFIC DNA-O-MATIC *
***********************

DNA (length=3): CAT

Original

S0: AAA
S1: CAT
S2: CAT

Reverse

S0: AAA
S1: TAC
S2: TAC

Reverse Reverse

S0: AAA
S1: CAT
S2: CAT

Reverse Complement

S0: TTT
S1: ATG
S2: ATG

Most Similar

Keyboard: CAT
File    : CAT
P-Stat  : 0.00
```

To complete this assignment you must submit the following:

1. An electronic copy of your program on Grace

a) Add a project named **02_DNAClasses** to your assignment solutions folder. It is vital that you name your solution and your project correctly!
b) Type your program (fully documented/commented) into the project. You need to follow the coding standards from the CS250 course page.
c) Pay attention to the example output. Your program's output must look **exactly** like the sample output. The spacing and newlines in your output must match exactly.
d) Make sure that your program builds without errors & warnings and runs correctly. If you get any errors or warnings, double check that you typed everything correctly. Be aware that C++ is case-sensitive. You will lose 10% if there are any warnings and 40% if your program does not build successfully.
e) Once you are sure that the program works, it is time to submit your program. You do this by logging on to Grace and placing your complete solution folder in the **CS250-0X Drop** folder.
f) The solution must be in the drop folder by the time class starts on the day the assignment is due. Anything submitted after that will be considered late.

2. A hard copy of your program

a) The hard copy must be placed on the instructor's desk by the time class starts on the day that it is due.
b) **The hard copy must be printed in color, double-sided, and stapled in the upper left corner if your solution contains multiple pages**. Failure to print properly will result in loss of 3.5 points (10%)
c) Your tab size must be set to 2 and you must not go past column 80 in your output.

Remember, if you have any problems, come to me straight away
with your project on Grace. Good Luck!!!! ☺