3.1 Description

Watson and Crick are most famously know for uncovering the structure of DNA (with contributions from Rosalind Franklin and Maurice Wilkins). They found that DNA is a 3D double helix structure made up of four nucleobases – Adenine (A), Guanine (G), Cytosine (C), and Thymine(T).

A growing branch of Computer Science(and genomics), is that of bioinformatics. This is the combination of using computer science techniques to analyze genetics data. In this assignment, we are going to learn some of the basics of bioinformatics using the programming abstractions we have learned.

3.2 Task

The tasks for this assignment are related to perform some operations on DNA.
1. First you will find the complimentary pair of DNA. You will be provided two test files(s1.txt, s2.txt) in which you will find the complimentary DNA strand.
   - You will open a file, and then use C++ to create a new file(s1_compliment.txt, s2_compliment.txt) and write the complimentary base pairs.
   - This means that for every 'A' you find in the file, you will output a 'T'. For every 'T' you find, you will output an 'A'. For every 'G', you will output a 'C'. For every 'C' you will output a 'G'. A biologist calls these pairs antiparallel.

2. Next, we will find the global alignment between s1.txt and s2.txt.
   - The following shows a perfect match
     
     | s1.txt | A | T | G | C | T | G | A | T | G | A | C | T |
     | s2.txt | + | + | + | + | + | + | + | + | + | + | + | + |
     
     For this table, you will output a score of 12, as there are 12 matches
   - The following shows a partial match
     
     | s1.txt | A | T | A | T | T | G | A | T | G | A | C | T |
     | s2.txt | + | + | x | x | + | + | + | + | + | + | + | + |
     
     For this table, you will output a score of 10, as there are 10 matches.
     Note to yourself, if one sequence is shorter than the other, can the score increase?

3. Finally we will be counting specific Amino Acid sequences that occur within the DNA strands s1.txt and s2.txt. This is the start of a process known as gene mapping.
   - For this, you will use the following reference table to find the 3 Stop codons. http://www.cbs.dtu.dk/courses/27619/codon.html
   - Each of the 3 stop codons consists of 3 of the DNA codons. So you must read three items in, and check if they are one of the stop codons.
   - Here is an example
     
     | A | T | A | T | T | G | A | T | G | T | A | A |
     
     - You will then output how many stop codons you found in the sequence.
3.3 Input Files
You will download the following files from the website for use with your project. Note that you will not provide these files when submitting.

Sequences
• s1.txt
• s2.txt

3.4 Files
You will be creating the files from scratch for this Assignment.
You will have files named exactly (including the exact capitalization):
• assignment3.cpp
• README

3.5 Submission

Listing 3.1: Submit Assignment

3.6 Expected Output
You can run a completed version of the program on by logging into the server and executing this file. /g/11/2017uc/solution/ps3

3.7 Evaluation/Rubric
• Files are properly named as specified above.
• You include a file called README containing the following information
  – Your name
  – Roughly how many hours you spent building the program
  – The command line you used to compile your program
  – The command line you used to run your program
  – Any known bugs in your program.
  – Any teaching assistants who assisted you or students you discussed the assignment with (if any).
• Part of your assignment will be graded based on code style (Did you use the function template or something more descriptive, did you add comments in your code, are variables well named, etc.).

• The remaining part of your assignment will be graded based on the correctness, and that it works to the specification.
  – Each of the three sections will be worth points.
  – Your program only needs to run once.
  – If you do anything creative or beyond the scope of the project, it should be documented accordingly.

3.7.1 Style

For this assignment, it is required that you write functions. When you write functions, you will want to document them using this template. An example is shown below:

```c
// Desc:
// Input:
// Output:
// Error Conditions:
int globalAlignment()
{
    Your code here
    return ...
}
```

Listing 3.2: Submit Assignment

3.7.2 Tips

• Try creating your own s1.txt and s2.txt that are much smaller than the given s1.txt and s2.txt. It will be easier to debug!

• Use print statements to debug, but remember to remove them later on.

• You can use tools as primitive as find in a text editor to verify your work on the provided DNA sequences, or ones you create on your own. Note that I will use different ones however for grading!