CS150 Assignment 7
DNA[1]

Date assigned: Friday, November 14, 2014
Date due: Monday, November 24, 2014, 1:00pm (40 points)

A DNA sequence is represented as a series of characters: A, C, T, G representing the nucleotides Adenine, Cytosine, Guanine, and Thymine. Certain regions of the DNA are called genes. Most genes encode instructions for building proteins (they're called "protein-coding" genes). These proteins are responsible for carrying out most of the life processes of the organism.

Nucleotides in a gene are organized into *codons*. Codons are groups of three nucleotides and are written as the first letters of their nucleotides (e.g., TAC or GGA). The sequences of DNA that encode proteins (the genes) occur between a *start codon* (which we will assume to be ATG) and a *stop codon* (which is any of TAA, TAG, or TGA).

For this assignment, you will write a complete C++ program that reads in sequences of genes from a file, and outputs a count of the individual nucleotides (A, C, G, T), a list of the codons, and a determination on whether the sequence is a protein-coding gene.

Input: The input file contains an integer representing the number of sequences in the file, followed by each of the sequences on a separate line.

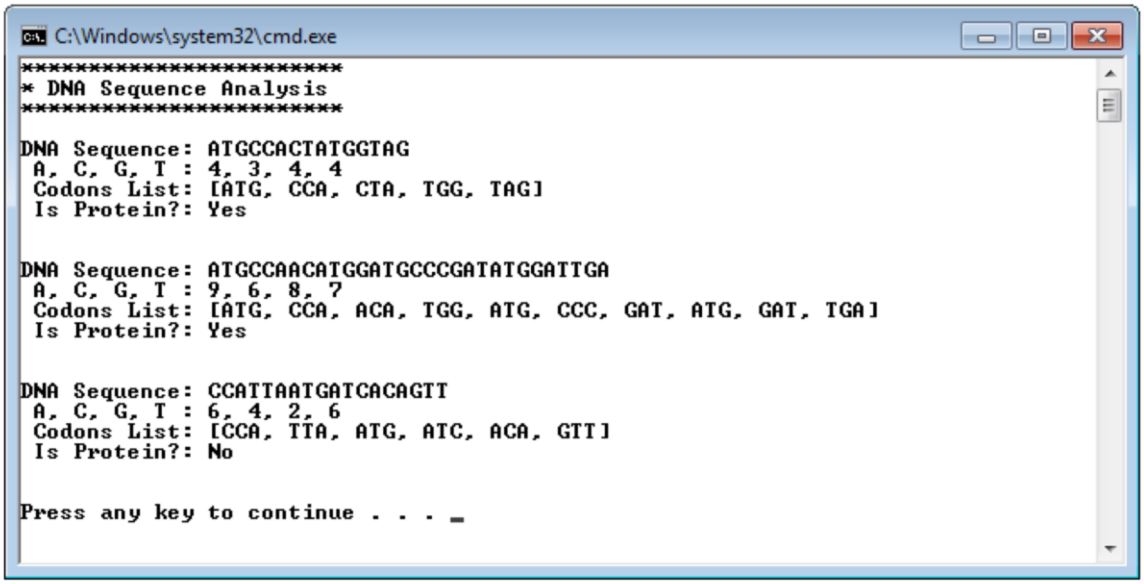
```
3
ATGCCACTATGGTAG
ATGCCAACATGGATGCCCGATATGGATTGA
CCATTAATGATCACAGTT
```

Output: For each sequence you are to output to the screen:

1. The sequence read in from the file
2. The count of each of the nucleotides (A, C, G, T)
3. A list of the codons. Codons are groupings of three nucleotides.
4. A determination if a sequence is a protein-coding gene. A protein-coding gene must:
    a. Begin with a valid start codon (ATG)
    b. End with a valid stop codon (TAA, TAG, or TGA)
    c. Contains at least five codons including the starting and stopping codon

[1] This assignment is based on an assignment used for the Java course at the University of Washington

Sample Run:



```
************************
* DNA Sequence Analysis
************************

DNA Sequence: ATGCCACTATGGTAG
 A, C, G, T : 4, 3, 4, 4
 Codons List: [ATG, CCA, CTA, TGG, TAG]
 Is Protein?: Yes


DNA Sequence: ATGCCAACATGGATGCCCGATATGGATTGA
 A, C, G, T : 9, 6, 8, 7
 Codons List: [ATG, CCA, ACA, TGG, ATG, CCC, GAT, ATG, GAT, TGA]
 Is Protein?: Yes


DNA Sequence: CCATTAATGATCACAGTT
 A, C, G, T : 6, 4, 2, 6
 Codons List: [CCA, TTA, ATG, ATC, ACA, GTT]
 Is Protein?: No


Press any key to continue . . . _
```

Notes:

- Assume that each sequence will contain a multiple of three nucleotides

- Assume that each sequence will only contain the valid uppercase characters A, C, G, T

- There will be a maximum of 500 nucleotides in an individual sequence

- You must use at least five functions in your program

- Code and test one function at a time.

- The debugger is your friend!

You must use the following functions:

- **`void writeTitle (string title);`**
  Prints the title passed to the function on to the screen.

- **`void readDNASequence (ifstream & inputFile, char sequence[], int max, int & size);`**
  Reads in one gene sequence from a file and puts it into the array sequence. max is the maximum size of the array (500), and size will contain the size of the actual sequence.

- **`void writeDNASequence (char sequence[], int size);`**
  Displays the gene sequence in array sequence to the screen. size is the number of nucleotides in the sequence.

- **`void countNucleotides (char sequence[], int sequenceSize, int counts[], int countSize);`**
  Calculates the number of each A, C, G, T and places the counts in the array counts. counts is a 4 element array, where element 0 stores the number of A, element 1 stores the number of C, etc.

- **`void writeCounts (int counts[], int size);`**
  Displays the counts to the screen.

- **`void writeCodons (char sequence [], int size);`**
  Displays the codons in the sequence to the screen.

- **`bool isProtein (char sequence [], int size);`**
  Returns true if the sequence is a protein, false otherwise.

You can create additional functions if you like.

To complete this assignment you must submit the following:

1. **An electronic copy of your program on Grace**
   a. Add a new project named **07_DNA** to your previously created assignment solution called PUNetIDAssignments. It is *vital* that you name your project correctly!
   b. Type your program (fully documented/commented) into the project. We are now commenting each function in a program. You must follow the coding standards!
   c. Pay attention to the example output! Your program's output must look **exactly** like the example output! The spacing and newlines in your output must match exactly.
   d. Make sure that your program compiles and runs correctly. If you get any errors, double check that you typed everything correctly.
   e. Make sure that your program does not produce any warnings.
   f. Once you are sure that the program works correctly it is time to submit your program. You do this by logging on to Grace and placing your complete solution folder in the **CS150-01 Drop** folder. This solution folder must contain seven projects.
   g. The program must be in the drop folder by 1:00pm on the day that it 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 1:00pm on the day that it is due.
   b. The hard copy must be printed in color, double-sided, and stapled if necessary.
   c. Your tab size must be set to 2 and you must not go past column 80 in your output.

Good luck! And remember, if you have any problems, come and see me straight away. ☺