Individual Programming Project Option 4:
DNA translation

A DNA string is a sequence of the bases a, c, g, and t in any order, whose length is usually a multiple of three. In reality, it is not necessarily a multiple of three, but we will simplify it as such for discussion. For example,

aacgtttgtaaccagaactgt

is a DNA string with a length 21 bases. Recall that a sequence of three consecutive letters is called a codon. Assuming the first codon starts at position 1, read *left-to-right*, and the codons are

aac, gtt, tgt, aac, cag, aac, and tgt

Those of you who know a little about genomics know that the open reading frame can be shifted to get a different set of codons. I want any of you who know this much to assume for discussion simplicity that there is only one open reading frame – the one starting at position 1.

With respect to the central dogma of molecular biology[1] and gene transcription[2], codons are the basic units of gene expression. Genes are initially expressed on a primary messenger RNA transcript (commonly recognized as ***mRNA)*** as a sequence of the codons, based on the ***DNA coding strand,*** while being made with the template strand.

Assuming the standard genetic code[3], when a protein macromolecule is ***translated*** from the coding strand transcript, each codon corresponds to one amino acid. That said, an amino acid might be represented by more than more than one codon. Three particular codons, TAA, TAG, or TGA, do not correspond to an amino acid at all. Instead, they are uniquely recognized as ***stop codons.***

[1] https://en.wikipedia.org/wiki/Central_dogma_of_molecular_biology
[2] https://en.wikipedia.org/wiki/Transcription_(biology)
[3] https://en.wikipedia.org/wiki/DNA_codon_table

Instructions

Write a Python program that will accept the pathname of dna file as it only command line argument.

Generally, the dna file should be a text file containing a valid DNA string with no newline characters or white space characters of any kind *within* it. (It will be terminated with a newline character.) This dna file should contain nothing but a sequence of the bases `a`, `c`, `g`, and `t` in any order.

Error checking:
The script should check that the dna file has only the letters `a`, `c`, `g`, and `t` and no other letter characters. If it does not satisfy this constraint, the script should output an appropriate user error message and then exit.

The script does not have to check that the file contains a number of characters equal to a multiple of three nor if the file ends with a newline character, where the number of characters is equal to a multiple of three plus 1. It does not have to satisfy this constraint, but for some extra credit for the project, the script could output an appropriate user error message and then exit.

The dna file is assumed to be the *coding strand* for a gene, reading left-to right. For simplicity of this discussion, assume the open reading frame begins at position 1.

The program must determine the amino acid sequence beginning with the first codon, all the way to the last available codon of the sequence. That is, if there are 2 remaining bases or 1 remaining base at the end of the sequence, these bases are discarded.

When reporting the amino acid of each codon, use the three-letter abbreviation except for the three possible stop codons (`TAA`, `TAG`, or `TGA`). When a stop codon is found, the program should report the stop codon as three asterisks: `'***'`.

The program must also report

- the counts of each amino acid determined from the sequence, sorted in order of decreasing frequency,
- the total number of different amino acids found, and
- if any stop codons were found.

Consider a file named dna_file contains the DNA string

```
acaatggtccctattagtgggcggcggcccgtataaact
```

For example, if the program is named dnaTranslate.py, and an end-user types in the command line

```
./dnaTranslate.py dna_file
```

The program should read the data in dna_file, and process the sequence within it. After text processing, the program should output the requested information.

```
$./dnaTranslate.py dna_file

Amino Acid Sequence: THR MET VAL PRO ILE SER GLY ARG ARG PRO VAL
*** THR

Amino Acid Counts:
ARG 2
PRO 2
THR 2
VAL 2
GLY 1
ILE 1
MET 1
SER 1

8 amino acids found.
A stop codon was found.
```

Important: If two or more amino acids have the same frequency, your script should break the tie using alphabetical order of the amino acids.

The program should not count a stop codon as an amino acid for the reports. If a stop codon was never found, the program should report that it did not find a stop codon instead.

Testing:
Use the DNA text files as test files for your program, located in the Linux Lab network in the cs132 course directory:

```
/data/biocs/b/student.accounts/cs132/data/dna_textfiles
```

The program should be thoroughly tested; you can create your own sample dna files based on the ones provided.