Individual Programming Project
Option 2: Gene Codon Sequence

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 of 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*, 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.

However, the *directionality* of the single DNA string has not been considered. If the string were to be read in the reverse direction, the assumption of *right-to-left,* the codons would instead be

`tgt`, `caa`, `gac`, `caa`, `tgt`, `ttg`, and `caa`

Those of you who know of the anti-parallel characteristics of DNA, please assume the directionality is not given to you and henceforth both directions are possible.

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. The mRNA sequence length is thus three times the number of codons**.**

Assuming the standard genetic code[3], the start codon, the first nucleotide base triplet, is commonly found to be ATG on the coding strand. The last codon, also known as the stop codon, is found to be one of the three possible codons: TAA, TAG, or TGA.

[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. If the file ends with a newline character, then the number of characters is equal to a multiple of three plus 1. If it does not satisfy this constraint, the script should output an appropriate user error message and then exit.

The dna file is assumed to be the *coding strand* for a gene, but the direction of the coding strand is not given.

- Both directions of the strand must be considered in your program.
- For simplicity, assume the open reading frame begins at position 1 for both directions.

The program must report the codon sequence beginning with the starting codon of transcription of the coding strand, '`atg`', and end in one of the three stop codons (`taa`, `tag`, or `tga`). The program must also report the number of codons (the sets of three bases), and the total number of bases that are in the transcript.

Consider a file named `dna_file` contains the DNA string

```
acaatggtccctattagtgggcggcggcccgtataaact
```

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

```
./geneCodonSeq.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.

```
$./geneCodonSeq.py dna_file

Forward Direction: atg gtc cct att agt ggg cgg cgg ccc gta taa
Number of Codons: 11
Number of bases: 33
Reverse: atg ccc ggc ggc ggg tga
Number of Codons: 6
Number of bases: 18
```

If the file does not contain a codon sequence for a gene as described (beginning with the start codon and ending in one of the three stop codons) in both directions of the dna file, it must print a message to the user instead stating that no sequence is found.

```
$./geneCodonSeq.py dna_file

No gene codon sequence found.
```

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 some sample dna files based on the ones provided.