

## Programming Exercise: Finding All Genes

### Assignment

This assignment is to find all the genes in a DNA string. We will specify the algorithm that defines a gene, which is different than the algorithm we used to find a single gene in the last lesson. You will start with code from the lesson and modify it to solve this problem. You should write and test each method on small data before moving on to the next step, finding all the genes in small strings of DNA to help you debug your program.

### Algorithm to identify multiple genes in a strand of DNA

1. To find the first gene
  - a. Find the start codon ATG.
  - b. Next look immediately past ATG for the first occurrence of each of the three stop codons TAG, TGA, and TAA.
  - c. If the length of the substring between ATG and any of these three stop codons is a multiple of three, then a candidate for a gene is the start codon through the end of the stop codon.
  - d. If there is more than one valid candidate, the smallest such string is the gene.  
The gene includes the start and stop codon.
2. If no start codon was found, then you are done.
3. If a start codon was found, but no gene was found, then start searching for another gene via the next occurrence of a start codon starting immediately after the start codon that didn't yield a gene.
4. If a gene was found, then start searching for the next gene immediately after this found gene.

**Specifically, to implement the algorithm, you should do the following.**

1. Write the method `findStopIndex` that has two parameters `dna` and `index`, where `dna` is a String of DNA and `index` is a position in the string. This method finds the first occurrence of each stop codon to the right of `index`. From those stop codons that are a multiple of three from `index`, it returns the smallest index position. It should return -1 if no stop codon was found and there is no such position. This method was discussed in one of the videos.
2. Write the void method `printAll` that has one parameter `dna`, a String of DNA. This method should print all the genes it finds in DNA. This method should repeatedly look for a gene, and if it finds one, print it and then look for another gene. This method should call `findStopIndex`. This method was also discussed in one of the videos.
3. Write the void method `testFinder` that will use the two small DNA example strings shown below. For each string, it should print the string, and then print the genes found in the string. Here is sample output for some test DNA strings:

DNA string is:

```
ATGAAATGAAAA
```

Gene found is:

```
ATGAAATGA
```

DNA string is:

```
ccatgccctaataaatgtctgtaatgtaga
```

Genes found are:

```
atgccctaa
```

```
atgtctgtaatgtag
```

DNA string is:

```
CATGTAATAGATGAATGACTGATAGATATGCTTGTATGCTATGAAAATGTGAAATGACCCA
```

Genes found are:

ATGTAA

ATGAATGACTGATAG

ATGCTATGA

ATGTGA