## Intro to CS / Python — Programming Assignment #4

Due: See Due Date in your Course Management System (25 points)

### Problem Statement – Putting it All Together

This program will open an input data file (a text file) provided by the user. On each line of the data file is a DNA fragment consisting of the character G,C,A,T. Your program will read each fragment and keep up with how many are in the data file. You must also write at least one meaningful function for this program.

Each DNA fragment has a sequence of G,C,A,T characters representing the bases in the fragment. We are going to generate a report showing the GCRatio of each fragment (see lab 5 for the definition of this value), whether or not the ratio falls into the 35%-65% range (most human fragments do), and if the fragment is too short to process (a fragment with fewer than 30 characters is not meaningful, and therefore should not be processed. Lowercase and uppercase characters should be treated as the same value.

Remember, when using an input data file, your file should exist and be stored in the same directory as your program file. Below is a short sample input data file and the resulting output file produced from this data set:

```
AAAGCCTATCGTTAAACCCGGTATATATCGGTATA
AAATTAGTTAAACAATGTATATATCGGTATAATCGTTATA
GCATCGGTATAAAAGCCTATCGTTAAACCCGGTATATATCGGTATAccc
TTA
```

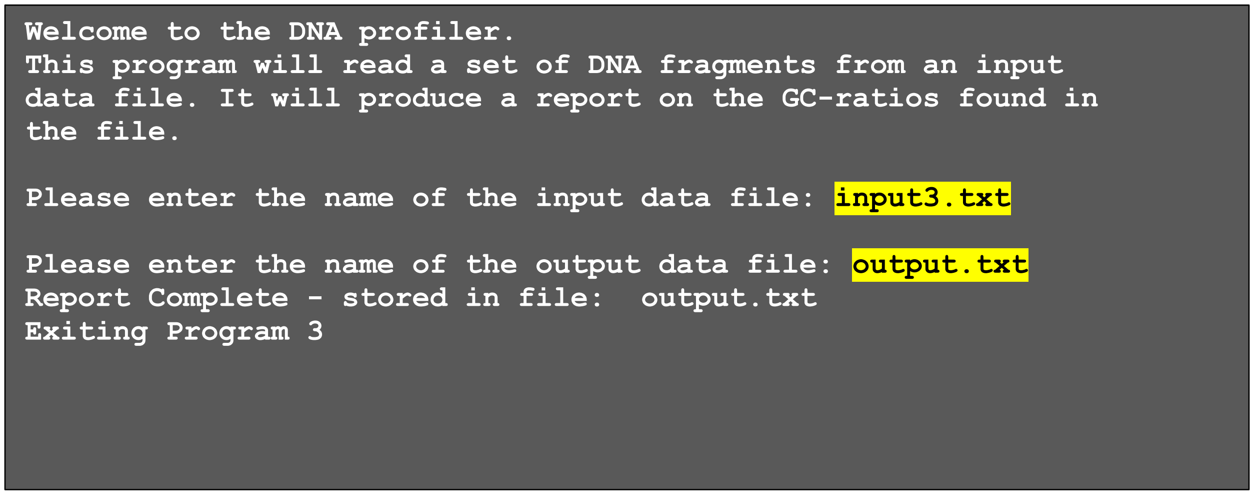
```
REPORT ON INPUT FILE:  input3.txt

               FRAGMENT                                    GCRatio    Other messages
-----------------------------------------------------------------------------------------------
AAAGCCTATCGTTAAACCCGGTATATATCGGTATA                 :   0.37  Fragment within the range 35% - 65%
AAATTAGTTAAACAATGTATATATCGGTATAATCGTTATA            :   0.20
GCATCGGTATAAAAGCCTATCGTTAAACCCGGTATATATCGGTATACCC :   0.43  Fragment within the range 35% - 65%
TTA                                                 :   Fragment is too short to process
----------------------------------------- SUMMARY -----------------------------------------------
There were 4 fragments found.
1 fragment was/were not long enough to process.
The longest fragment found had 49 values
```

*NOTE: To make formatting easier, you may assume that no fragment is longer than 50 characters.*

## Sample Execution

Below is shown a screen capture of what the program would look like if I ran it with an input file with the name **input3.txt** that is stored in the same project directory with my program. The values in **YELLOW** are what were entered by the user in this example execution.



```
Welcome to the DNA profiler.
This program will read a set of DNA fragments from an input
data file. It will produce a report on the GC-ratios found in
the file.

Please enter the name of the input data file: input3.txt

Please enter the name of the output data file: output.txt
Report Complete - stored in file:  output.txt
Exiting Program 3
```

**HINTS/SUGGESTIONS:**

- Get your report working for a single DNA fragment.
- Then add the logic to read each line from the input file and process it. Add your counters and the logic needed when you have the file processing correctly.
- Have the data write to your screen. Once you are satisfied with it, add the changes to have it print to an output file instead of the screen.

## Turn In:

Include a copy of the input file(s) you tested your program with. I will test your program with many input files, so do not hardcode the filename into your program! Prompt the user for it as in the example.

## Grading Requirements

- Your program must be well-commented. Comment all variables, and remember to have a section of comments at the top of your program that includes your name, date, course section and a description of what your program does. Refer to my example solutions and the textbook for good style in comments.
- Use meaningful variable names.
- Format your output to the output file so it appears like mine in the report file.
- You will receive no more than 50% credit if your program does not compile or crashes.
- If your program compiles but does not execute correctly, you will receive no more than 70% credit. Test your program with more data than just my sample (i.e. create your own set of different input files to use).