PROGRAM 04-1 WORKING WITH FILES AND STRINGS

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 characters G,C,A,T. Your program will read each fragment and keep up with how many are in the data file.

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, 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.

Given a single string as input where each character is one of **G, C, T or A,** the GC ratio is computed as the count of G+C values over the total count of A,T,C,G values. It is often written in texts as the following, where each letter represents the count of instances in the string:

$$\frac{G + C}{A + T + G + C}$$

Remember, when using an input data file, your file needs to be created and 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
```

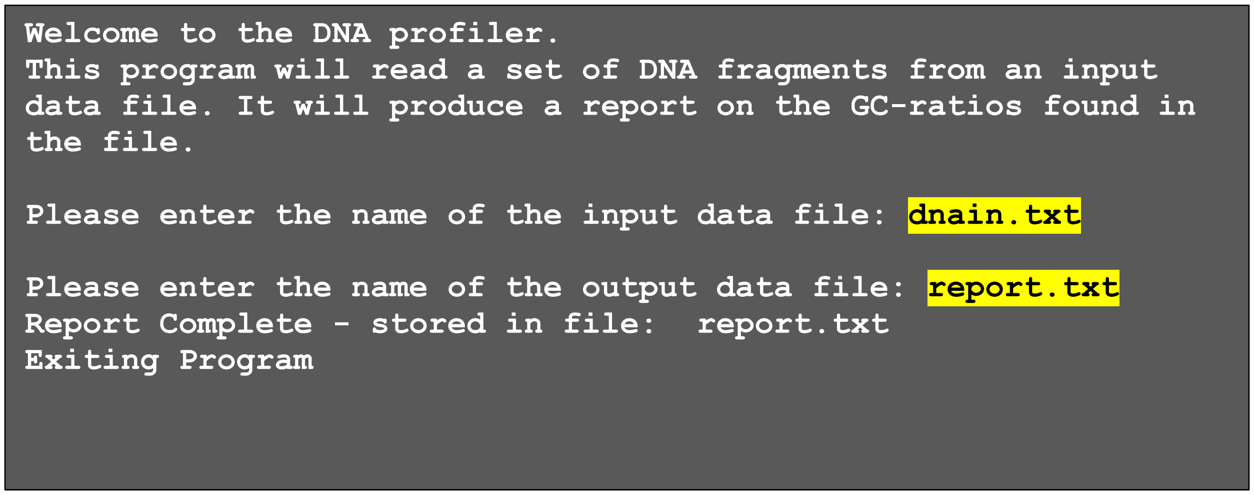
```
REPORT ON INPUT FILE:  dnain.txt

               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(s) were not long enough to process.
```

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

You may assume that there are no bad characters in the sequence.

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 **dnain.txt** that is stored in the same project directory with my program.exe. (Or you may store it elsewhere, and type the enter path as part of the filename). 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: dnain.txt

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

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.
- You will need to #include <string> to use the string data type

GRADING NOTES:

- In addition to the commenting, and programming style requirements mentioned in earlier assignments, you should also begin paying attention to efficiency of code. If you find yourself copying and pasting the same logic in multiple places, it may need to be restructured so that you only need to write the code a single time. Perhaps the only difference is that one version adds a partial text message to the end of a line of output. It is not necessary to copy all of that logic for producing the first part, again.
- I will test your program with many input files so please do not hardcode the filename into your program! Prompt the user for it as in the example.

TURN IN:

Along with your program file, upload a copy of the input file(s) you tested your program with.