

CS105 : Computer Programming : PERL

Project 01

Assigned on Wed, Feb 7
Due on Wed, Feb 14 at 11:59PM

1 Objectives

For two weeks, we have learned some of essential concepts required to write programs in Perl. Through this project, we want to review and practise the concepts by incrementally modifying the provided code (shown in next section). We start with a simple program code that only contains a basic program structure. However, gradual modification will result in a final program that has advanced functionality. For example, it will get a file name from a commandline, read in some data from a text file, store data in a hash, perform a simple calculation, and print the results on a screen and a text file.

2 Description

For this project, first download the code (“computeFrequency.pl”) and the data file (“sequence.txt”) from the class webpage.

Make sure your program uses the following perl pragmas:

- use warnings;
- use strict;

This program does the followings:

- Open to read a text file called “sequence.txt” (using “open()”)
- Get rid of the new line character from the line just read (using “chomp()”)
- Get the length of the line (using “length()”)
- Get each character in the line (using “substr()”)
- Count frequency of each character (using hash)
- Sum all the line lengths to get a total length
- Close the file handle (using “close()”)
- Get keys (using “keys()”)
- Print results (using “print()”)

3 Grading Criteria & Policy

- Make sure the data file (“sequence.txt”) is in the same directory where you run the code (i.e., `>perl computeFrequency.pl`). Is it working as you understand and expect? (**No point for this**)

Now, we get started to modify the code and the modified code will be called “computeFrequency1.pl”.

1. Modify the code so that we can run the code by just calling the filename as follows (See p.13 and 33, **(5 points)**):
`>./computeFrequency.pl`
2. Modify the code so that it can take the data file name from the keyboard as follows (See p.51, 52, and 68 **(5 points)**):
`>Type data filename:`
3. Modify the code so that it prints out the count of each character as follows (See p.91, 95, and 97 **(10 points)**):
`>The character A occurs 1142136 times.`
4. Modify the code so that it prints out the total character length of the data as follows (See p.30 **(5 points)**):
`The total number of characters is 4639221.`
5. Modify the code so that it computes the fraction of each character. (For example, $\text{fractionOfA} = (\text{countOfA} / \text{totalLength}) * 100$) Also, print out the value as follows (See p.30 and 91, **(10 points)**) :
`>The character A occurs 1142136 times, or 24.6191332553461 %.`
6. Modify the code so that it prints out each character in an alphabetical order as follows (See p.97 **(5 points)**) :
`>The character A occurs 1142136 times, or 24.6191332553461 %.`
`>The character C occurs 1179433 times, or 25.4230828839583 %.`
7. Modify the code so that it takes the result (i.e., output) filename from the keyboard as follows (See p.51, 52, and 68 **(5 points)**):
`>Type result filename:`
8. Modify the code so that it stores the followings to the result file, which will be called “result1.txt” **(5 points)**:
Name = Your name
UT EID = Your eid

Data file = sequence.txt
Perl Program = computeSequence1.txt
Result file = result1.txt

The character A occurs 1142136 times, or 24.6191332553461 %.

...

The total number of characters is 4639221.

End of file.

The following is a challenging problem. However, it is worth trying to double-check your understanding of PERL programming.

- **(Challenge problem, (50 points))** Modify the code so that it compute frequencies of all possible two-character words. The code will be called “computeFrequency2.pl” and the result file name is “result2.txt”. The contents of “result2.txt” should be as follows:

Name = Your name

UT EID = Your eid

Perl Program = computeSequence2.txt

Data file = sequence.txt

Result file = result2.txt

The two-character word AC occurs 256658 times, or 5.5323524213122 ...

The total number of characters is 4639221 - 1.

End of file.

- Please make sure to include the followings in both the code (“computeFrequency1.pl” and “computeFrequency2.pl”):

```
# Project = 01 (or 02)
```

```
# Description = computes frequency of a character (or two-word character)
```

```
# Program = computeFrequency1.pl (or computeFrequency2.pl)
```

```
# Input = sequence.txt
```

```
# Output = result1.txt (or result2.txt)
```

```
# Usage = >./computeFrequency1.pl (or >./computeFrequency2.pl)
```

```
# Name = Your name
```

```
# UT EID = Your eid
```

```
# Comments (or README) = Describe some thing that you wish the grader to know.  
For example, What do you like/dislike about the project? What was your challenge  
for this project?
```

```
Perl code will be followed.
```

- This project is worth a total of 100 points.
- You will need to submit two files for this project: “computeFrequency1.pl” and “computeFrequency2.pl” Submit the files as follows:
>turnin -submit hyukcho project01 computeFrequency1.pl computeFrequency2.pl

Good luck!