**BIFS 617: Advanced Bioinformatics**

**Final Exam**

**Dr. Alkharouf, UMUC**

**Deliverable and Instructions (please READ!):**

This is an individual exam, you may not seek help or offer help to anyone!

For each of these questions submit the .pl file, name the files based on the question they belong to, so Q1.pl, Q2.pl…etc. (i.e. no need to include your name! I will be creating separate folders for each of you).

Also copy and paste all your code into a Word document, this I will submit to Turnitin.com.

Please put everything in one folder, **compress it** into a zip file and submit that one zip file.

For questions 8 and 9, use at least one subroutine when solving them. For Q8 and 9 you may use any module or subroutine from the book(s) or CPAN. For all other questions, no pre-existing modules or subroutines are allowed.

**Cite your references INSIDE your code (as comments). Any code used that is not yours and is not cited will be considered plagiarized and you will be given a zero on the test.**

**1.** (10 pts) Write a Perl program that given a DNA string, prints out the 20 characters upstream of the start codon ATG. That is, given:  
  
$dna = "CCCCATAGAGATAGAGATAGAGAACCCCGCGCGCTCGCATGGGG";  
  
print out:  
  
The 20 bases upstream of ATG are AGAGAACCCCGCGCGCTCGC  
  
**Use a regular expression to match the desired substring**.

**2.** (10 pts) Write a Perl subroutine that reads in a file containing two strings on each line, and creates a hash with the first string as key and second string as value. Test your subroutine on a file containing the following lines (copy the text and paste it in notepad, and then save it). Your code should work with any size file, not just the one given!  
  
color blue  
shape round  
weight 150  
speed fast

**3.** (10 pts) Write a program that will predict the size of a population of organisms. The program should ask for the starting number of organisms, their average daily population increase (as a percentage), and the number of days they will multiply. For example, a population might begin with two organisms, have an average daily increase of 50 percent, and will be allowed to multiply for seven days. The program should use a loop to display the size of the population for each day. So for the previous example, the output should look like:

Day Organisms

-----------------------------

1 2.0

2 3.0

3 4.5

4 6.75

5 10.125

6 15.1875  
7 22.78125

**4.** (10 pts) Write a Perl program that adds up the numbers in a file and prints out their sum, average, max and min. Assume that there is one number per line. Print the average out showing two digits after the decimal point (Hint: look up the printf command).

Test your program with a file containing:

40

10

2

3

4

Your output should look like:

sum = 59

ave = 11.80

max = 40

min = 2

Your code should work on **any file** containing different or more numbers than listed above (i.e. don’t just assume you have 5 numbers in a file).

**5.** (10 pts) Write a Perl script to compute the average for **each column** of numbers in a file with the following format:  
  
1 2 3  
5 4 6  
0 2 4  
etc.  
  
The data file may have any number of rows, but will always have 3 columns.

**6.** (10 pts) Write a Perl script to print out the GI numbers from each header **in a FASTA file** of sequences. Assume that the headers of the form:  
  
>gi|1234567| more info ..

Go to GenBank and download a set of 5-10 FASTA sequences to test your code.

**7.** (10 pts) Modify the code in the lecture notes (and book) so that it parses the DNASIS restriction enzyme file (see attached – this is just a small sample of the file so you can test your code on) instead of the BIONET file (which was used in the lecture notes and book).

Hint: You only need to make small changes! The goal here is to see if you understood the code, you need to fully understand the code in the lecture notes and book before attempting to modify it to work for the new file.

**8.** (15 pts) Next generation sequencing is used to sequence RNA samples to get accurate measurements of gene expression on a genomic scale. Write a Perl program that parses out the attached sequence read alignment file (6\_perianth\_A\_filtered.SAM) to count how many reads a gene produced (a higher number indicates a gene that is highly expressed). You basically just have to count the number of times a gene ID (like gene29004) occurs in the given sequence. All lines starting with @ are comment lines and should be ignored. Print out the gene ID’s and their counts once done. For instance if you find gene29004 mentioned 3 times while gene23457 6 times in the file, the output should be:

Gene ID: Number of reads aligning:

gene29004 3  
gene23457 6

**9.** (15 pts) Design a Perl program that takes the following DNA sequence file (test\_seq.txt – see attached) and mutates it while maintaining the same base pair distribution (i.e. shuffles the base pairs). Once mutated\shuffled, find the similarity between the mutated and original DNA by calculating a score based on the following criteria:

If a purine was mutated to another purine --> -1

If a pyrimidine was mutated to a pyrimidine --> -1

If a purine was mutated to a pyrimidine or vise versa --> -2

If no change occurred --> 0

An example of how the score is calculated is shown below:

Original: AGCCGTAGCT

Mutated: AATGTACGAT  
Score = 0-1-1-2-2-2-2+0-2+0 = -3

Note: A and G are purines while C and T are pyrimidines.

Print out the score for the user to see. Also print out the original and mutated sequences aligned base pair by base pair (i.e. the way Blast reports do it).

Note: Accept the file as a command line argument. You will also have to remove the comment line “>…etc.” before storing the sequence in your program.