Here are some exercises selected or adapted from Rosalind (http://rosalind.info), an online collection of Bioinformatics problems for beginners. The first problem is required, while the other two are optional for practicing in your programming language of your choice.

For each problem, write a program that outputs the solution. For this homework and for the rest of the course in general, we highly encourage you to use Julia to implement your solutions.

The TAs have written a Julia tutorial for CSCI1810: cs.brown.edu/courses/csci1810/juliatutorial.pdf.

To hand in, log into a department linux machine. Put all of the files that you want the TAs to see into a directory. Then navigate to that directory and type `cs181_handin warmup`. This will recursively hand in your entire directory. You should receive an email confirmation of your handin.

**Specifications:** To facilitate anonymized & automated grading, each of your solutions must be accompanied by a shell script. Make sure each problem is able to output the correct result, using the shell script provided. Each problem has a specification section that describes how it should be run. Also make sure your code works on the department machines.

To grab the support code, run the command `cs181_setup warmup`. For this project, we will simply provide you with sample shell scripts. For future projects, the setup command will also provide you with support/stencil code.

**Problem 1: Reverse Complement**

(Rosalind: REVC) In DNA, nucleotides A and T from opposite strands form a Watson-Crick base pair, as do nucleotides C and G. We say that A and T are complementary nucleotides, and C and G are complementary nucleotides. By convention, the string of nucleotides representing one strand of a DNA molecule is written in the $5' \to 3'$ direction, the direction of DNA synthesis. Thus, the string representing the other (complementary strand) is the reverse complement string.

The reverse complement of a DNA string $s$ is the string $s^C$ formed by reversing the symbols of $s$, then taking the complement of each symbol.

For example, the reverse complement of GTCA is TGAC.

**Given:** A DNA string $s$. 
Return: The reverse complement $s^C$ of $s$.

Specification: sh reverse.sh STRING

```
$> sh reverse.sh GTCA
TGAC
```

Note: Your implementation must not contain any usage of a built-in reverse() function on strings.

Problem 2: (OPTIONAL BONUS) Hamming Distance

(Rosalind: HAMM) Given two strings $s$ and $t$ of equal length, the Hamming Distance between $s$ and $t$, denoted $d_H(s, t)$, is the number of positions where $s$ and $t$ differ.

For example, the hamming distance between the pair of strings GAGCCTACTAACGGGAT CATCGTAATGACGGCCT is 7.

Given: Two DNA strings $s$ and $t$ of equal length.

Return: The hamming distance $d_H(s, t)$.

Specification: sh hamming.sh STRING1 STRING2

```
$> sh hamming.sh GAGCCTACTAACGGGAT CATCGTAATGACGGCCT
7
```

Problem 3: (OPTIONAL BONUS) Fibonacci Sequence

(Rosalind: FIBB) The Fibonacci sequence is a sequence of numbers $F_n$ for $n = 1, 2, \ldots$ With the exception of the first two terms, they satisfy the recurrence relation

$$F_n = F_{n-1} + F_{n-2}$$

Traditionally, $F_1 = F_2 = 1$, so the first 5 terms are 1, 1, 2, 3 and 5.

Use a dynamic programming approach to compute the $n$th fibonacci number.

Given: $n$, a positive integer.

Return: The $n$th fibonacci number, $F_n$.

Specification: sh fib.sh N

```
$> sh fib.sh 20
6765
```

Note: Do not submit a recursive solution, with or without memoization. Instead, use dynamic programming.