**Week 5 Practice Problems**

**#1.** Read one string from the user. Then, swap cases and print the result. In other words, convert all lowercase letters to uppercase letters and vice versa. You should solve this question twice.

1. There is a str method that will do this for you and so after the input just call it.
2. But a) is not much of an exercise, so you should also do it without calling that function!

**#2.** Get one string from the user, remove all exclamation and question marks from it, and output the resulting string.

Example:

Input: Hello!

Output: Hello

Input: How are you?

Output: How are you

1. There is a str method that will do this for you (if you use it correctly) and so after the input just call it.
2. But a) is not much of an exercise, so you should also do it without calling that function!

**#3.**

a) Prompt the user to enter their full name and, using only string comparison functions, decide if the string starts with your first name. If it does, print out “How did you know my name?”. If it doesn’t, print out “It’s rude to not know my name.”

b) Write a function that takes two strings and returns True if the two “match”, where this time “matching” is not case sensitive. (So “Smith” and “smitH” match). Redo part a) using this function.

**#4.** Redo #2 from Week 4 using strings. That is, write a function that takes in a positive integer of any size and returns the “flipped” version (that is, with its digits reversed). For example:

Enter a positive number: **90177**

That number flipped is 77109

Use string functions to do it. Note that the function should still have a type contract of (int)->int. It is only inside the function that you are allowed to use strings.

**#5.** DNA has 4 nucleotides: adenine (A), thymine (T), cytosine (C), and guanine (G). A strand of DNA can be represented as a string of A, T, C, and G. A double helix contains two strands that are "mirror images" of each other in the sense that A is swapped with T and C is swapped with G. For example, here are two paired strands:

ATGGGCAATCGATGGCCTAATCTCTCTAAG

TACCCGTTAGCTACCGGATTAGAGAGATTC

Notice that each A in the top strand has a corresponding T in the bottom one and vice versa. Same for C and G.

Write a function that takes in a DNA strand and produces its matching counterpart in a double helix.

There are a few ways to solve this problem see how many you can come up with.

Hint: You are going to have to be a bit clever with how you use the methods to get this right.

**#6.** Write a program that creates a copy of grades.txt, by reading each grade from the original file, and then writing that value to a file of a different name.

We’ll be looking at a number of different ways that this problem can be solved (see Ch 10 in your textbook). You should write three functions each solving the problem using different methods for reading the file. (Don't worry about readlines() yet).

**#7.** Open the file onetwo.txt with a simple text editor (e.g., Notepad on Windows or TextEdit on a Mac), and you’ll discover a sentence, but with each word on a separate line:

One,

two,

buckle

my

shoe.

Remember that although you can’t see them, the file contains newline characters at the end of each line. Write a program that reads the input file, and writes the words to the screen, all on one line; for example:

One, two, buckle my shoe.

As with Q6, you should write a number of functions each taking in the filename and returning the one-line string. Each function should solve the problem using a different method for reading the file.

**#8**: In fluid dynamics, the dimensionless Iribarren Number defined below is a parameter used to model waves on beaches and coastal structures.

**Iribarren Number =**

where α is the bed slope (in radians), *H* is the breaking wave height (in meters) at the edge of the surf zone, and *L* is the water wavelength (in meters).

Depending on the Iribarren number, the breaking waves are classified into *spilling*, *plunging*, and *collapsing or surging* as follows

|  |  |
| --- | --- |
| **Wave type** | **Iribarren Number (IR) number** |
| surging or collapsing | 3.3 < IR |
| plunging | 0.5 < IR <= 3.3 |
| spilling | IR <= 0.5 |

Write a function which takes as input parameter a string containing the bed slope, the wave height and the water wavelength as a string in the following format:   
“number1,number2,number3”, e.g., “1.2,2.3,5.7” and returns a string indicating the type of wave: "surging or collapsing", "plunging" and "spilling".

You can assume that:

* the input string does not contain any white spaces
* the three numbers are separated by commas
* the three numbers are in their correct units
* each of the three numbers may have any number of digits. The following are all valid inputs: “1,2,3”, “1.245,5.60,23”, “1.0,2.3456789,2.19”