Homework #1

For all homework assignments, you are free to consult others on concepts, but the final code you turn in should be your own. If you are just learning programming for the first time, I would suggest that you try to spend at least 30 min thinking about each problem before seeking assistance. Even then, the first few assignments may be very frustrating and time consuming, but if you don't practice the fundamentals now, you may be in real trouble later in the class. The only way to learn programming is by doing it. There are many possible solutions to each of these problems. If you need help, you can contact the TA or email me at any time, or find me any time my office door is open (mornings are usually better). We will go over the solutions at the beginning of class each Friday, so the homework must be emailed to the TA before then !

To hand in your homework: For each problem, create a ".py" file containing the program that solves it. This should not be a Word doc, or a PDF, but a text file you could execute directly. Use comments to document your programs!

Submit homework by email with the subject "Homework N” to koire@bcm.edu with a cc to sludtke@bcm.edu.

Practice #1

The homework (next page) is due next Friday. Before the Lab next Monday, it would be a good idea to get a little practice with data types, conversion and manipulation. These small exercises should not be turned in, but I strongly encourage you try and do them yourself over the weekend to gain a little experience. Set aside an hour and try to solve each one for ~10 minutes. After trying yourself, download the solutions and make sure you understand them all:

1) Create a list of numbers from 5 to 15 inclusive stepping by 0.5.

2) Start with the string "this is a short test string" and create a new string with the letters sorted alphabetically

3) Create a string containing only the unique letters in "abracadabra"

4) Start with s="1 2 4 8 16". Convert the string to a list of integers and take the log base 2 of each number.

Homework #1

Ask the user to enter a 1-letter DNA sequence, for example “CTGGGCCACACTGGAAGAACTGTGTTGGGCCACA"

Count the number of each nucleotide present in the entered sequence (and print the count)

Print the reverse complement of the entered sequence

Write a program to identify the winner of a rock,paper,scissors game. Ask the user what player 1 picked (rock, paper or scissors), then ask what player 2 picked. Finally, print the winner (player 1, 2 or tie)