Advanced Exercises

(optional - but then again, all of these are optional)

Exercise 1

I posted solution to all the previous exercises.

Go through my solutions and compare to yours.

Experiment with some of the alternatives I show.

Let me know of any mistakes.
(I know there's at least one.)

#2, FASTA files

I put a FASTA file at /coursehome/dalke/ls_orchid.fasta
(It’s part of the Biopython regression suite.)

Take a look at the file using `more` (or `less`).
Can you figure out the file format?

FASTA format

- A FASTA file contains 0 or more records
- Each record starts with a header line
- The first character of the header is a “>”
- After that are a bunch of sequence lines
- After the sequence lines is a blanks line
    - (In real FASTA files that line is optional. Don’t worry about that, since this version is easier to parse.)

Exercise #3

Write a program to count the number of records in a FASTA file.

Once done, modify that program to print the total number of bases present and the total count of each base found.

Exercise #4

Write a program to print only the header lines in that FASTA file. It must not print the leading “>”.

Exercise #5

Write a program to find the record(s) where the header line contains the text `P.tonsum`. Print it out, in FASTA format. (That is, the output should not be changed.)

Command-line arguments

In Python you can get the list of command-line arguments with sys.argv. This is a normal Python list.

```
> cat print_args.py
import sys
print "Command-line arguments are:", sys.argv

> python print_args.py
Command-line arguments are: ['print_args.py']
> python print_args.py ls_orchid.fasta
Command-line arguments are: ['print_args.py', 'ls_orchid.fasta']
> python print_args.py *.seq
Command-line arguments are: ['print_args.py', '10_sequences.seq',
'ambiguous_sequences.seq', 'many_sequences.seq', 'sequences.seq']
>
```

Exercise #6

Write a program which gets two command line arguments, turns them into floats, and prints the sum.

Note that sys.argv[0] is always the name of the Python program so you'll need to use [1] and [2]

You can find this program in the worked out solutions, under the first Monday. Compare it to yours.

Exercise #7

Modify your solution from #5 so that it uses the first user argument (sys.argv[0]) as the text to search for in the header. Have it print all FASTA records which contain that text. A good test case is "`P.`"

Exercise #8

Write a program to read a FASTA file and print the header line (without the leading ">") if and only if the sequence ends with a C. Use a command-line argument to get the filename. Test it against both FASTA files in my directory.