CSE 620: Project 1
Due: Sept. 6, 2013 2:30 PM

***Submission**:* In your SVN *Proj1* directory there is a file **dnaSeq.py**. All code for this assignment will be added to that file.

Purpose: We need a sequence class. Sequences are strings, so we could just use the string class. But there are two drawbacks:

1) Strings are immutable, and it will be useful to have a mutable string class.
2) We would like the option of implementing methods appropriate for a sequence class.

To this end we are going to write a mutable sequence class, implementing ten methods (seven "magic methods" used to support necessary functions, such as the "__len__(self)" method to support the **len** function, and three mutator methods), and two standalone functions for I/O.

Go through the **dnaSeq.py** file, and for each method uncomment the function header and implement the function. *Most of these should be very short.* Make sure to follow specifications in terms of raising errors as needed – especially in the magic methods, where the built-in Python methods count on getting the right kind of exceptions.

Most of the methods are self explanatory – only a few require further explanation.

__getitem__(self, i): This is used by the [] operator. The value *i* can be one of two types. (Anything else should results in raising a *TypeError*.)

- If *i* is an **int**, then we treat this is a normal index. In this case, you want to return the i^{th} character of the sequence. Your implementation should support negative indexing, and throw a **KeyError** if *abs(i) ≥ len(seq)*.
- The alternative is that *i* is a *splice object* – the object you get if the user calls *seq[i:j:k]*. In this case, you should be returning a *dnaSeq* object as appropriate. [1]

__setitem__(self, i, value): Needed to allow assignment through the [] operator. See comments on __getitem__.

fasta format: The I/O functions you write will assume *fasta format.* Fasta format is a standard, very simple, format for recoding biological sequence. A file in fasta format contains one or more sequences, such that for each sequence:

- There is a single line description of the sequence, starting with a > symbol.

[1] If you use the right data structure for storing the sequence information, this will be easy. You will just pass the slice object on to that and let it do all the work.

- The sequence is broken over the following lines, with *w* characters per line (with the last line possibly being less than *w* characters). *w* may vary from file to file, but should be consistent within the.
- The end of a sequence is followed by either: the description of another sequence (indicated by a ">" symbol); one or more blank lines (followed by the description of a sequence); the end of the file.

By convention, a the name of a fasta file always has either a **.fa** for **.fasta** extension.