Programming Project #04

Assignment Overview
This assignment will give you more experience with while loops and if statements, and give you an opportunity to work with strings. Project is due Monday Feb 4th before midnight.

The Problem
Scientists measure how closely related a species is by looking at the DNA sequences for key proteins and seeing how similar/dissimilar they are. If the two sequences of DNA are essentially the same, the two species are considered to be evolutionarily closer since there is a relationship between changes and time. This process is called sequence alignment.

Consider the two strings of DNA below (completely made up, miss matches in red):

Species 1: AATAACGAAA
Species 2: AAAACGAAAA

A scientist can change the alignment by assuming that an insertion or deletion, of one of the bases has occurred. They could make such a change, called an **indel** for short, to see if it improves the alignment:

Species 1: AATAACGAAA-
Species 2: AA-AACGAAAA

Assuming two indels, marked as two dashes(-), the alignment is greatly improved. The scientist would assume that two changes happened, one change in each species.

While complex algorithms exist to do sequence alignment, it is also useful to support a researcher and allow them to do an alignment by hand.

Program Specifications::
1. You will prompt for two strings. The strings can have any characters you like, but to be "biological" it should consist of: "A", "T", "C", "G". The strings **do not** have to be of the same length.
2. You will then prompt for one of 3 commands:
    a. "a" for add. Add an indel
    b. "d" for delete. Delete an indel
    c. "s" for score. Score the present alignment
    d. "q" for quit. Stop the process.
3. **Adding an Indel**. When you add an indel, you must prompt for two pieces of information:
    a. which string to change
    b. at what index (starting at 0) do you wish to place the indel (placement is **before** the given index, **<u>Error</u>** if the index is out of range).

The string should then be modified and a dash(-) added.
4. **Delete an Indel**: If you can add an indel, you should be able to delete it if it doesn't do what you want. Again, you must prompt for two pieces of information
   a. which string to change
   b. the index (starting at 0) to delete the indel. It is an **<u>Error</u>** to delete a character that is not an indel.
5. **Scoring**. You will report the number of matches and the number of mismatches.
   a. Any indel is automatically a mismatch.
   b. If one string is shorter than the other, the shorter string is filled out with indels.
   c. After you score, you print both strings.
        i. Matching characters are printed in **lower case**. If the user entered upper case letters, you convert them to lower case on a match.
        ii. All mismatches are printed in **upper case**.
        iii. Indels are printed as dashes.

Deliverables
Turn in proj04.py
1. Please be sure to use the specified file name, i.e. "proj08.py" (no main function!!!)
2. Save a copy of your file in your CSE account disk space (H drive on CSE computers).
3. Submit the files using the "handin" program: http://www.cse.msu.edu/handin/webclient

Assignment Notes:
1. As before, try to do this in pieces:
        a. get the command loop working
        b. get individual commands to work (scoring is the most work, save it till last)
2. Printing the string is best done by creating a new string and adding characters (upper, lower or dashes) to that string and then printing it. Then clear it before the next printing.

Example Run
```
String 1:aaabbbccc
String 2:aabbbccc
What do you want to do:
                a (add indel)
                d (delete indel)
                s (score)
                q (quit): s

Matches: 6 Mismatches: 3
Str1: aaAbbBccC
Str2: aaBbbCcc-

What do you want to do:
            a (add)
            d (delete)
            s (score)
            q (quit): a
Work on which string (1 or 2):2
```

```
Before what index:2

What do you want to do:
            a (add)
            d (delete)
            s (score)
            q (quit): s

Matches: 8 Mismatches: 1
Str1: aaAbbbccc
Str2: aa-bbbccc

What do you want to do:
            a (add)
            d (delete)
            s (score)
            q (quit): d
Work on which string (1 or 2):2
Delete what index:2

What do you want to do:
            a (add)
            d (delete)
            s (score)
            q (quit): s

Matches: 6 Mismatches: 3
Str1: aaAbbBccC
Str2: aaBbbCcc-

What do you want to do:
            a (add)
            d (delete)
            s (score)
            q (quit): q
```