CPS363: Introduction to Bioinformatics
Homework 2: Pairwise Sequence Alignment
(30 Points)

Handed out: February 12 (Friday)
Due: 11:59pm February 19, 2019 (Friday)
Late submissions accepted with penalty until 11:59pm February 20 (Saturday).

In this homework you will implement pairwise sequence alignment algorithms. The program should run in the terminal of Linux/Mac system. Your program should have the following options:

- It can perform pairwise sequence alignment of two protein sequences , or two DNA sequences depending on the user's configuration of parameters.
- It can perform global sequence alignment or semi-global sequence alignment depending on the user's configuration.
- It can choose the corresponding scoring matrix (dnaMatrix or BLOSUM45) file depending on if it is a protein sequence alignment or DNA sequence alignment.

Here are possible parameters which can be set for your program:

-i	First sequence [File In and should be FASTA format]
-j	Second sequence [File In and should be FASTA format]
-p	Align protein sequences or DNA sequences [T/F]
-atype	global, or semi-global [G/S]
-o	alignment output file [File Out]

Here are some possible ways of running your program (assume your program is called *align.py*):

- $*python3 align.py -i seq1.txt -j seq2.txt -o out.txt -p F -atype G*
  Perform global pairwise sequence alignment of two DNA sequences in seq1.txt and seq2.txt. The output will be in out.txt.
- $*python3 align.py -i seq1.txt -j seq2.txt -o out.txt -p T -atype S*
  Perform semi-global pairwise sequence alignment of two protein sequences in seq1.txt and seq2.txt. The output will be in out.txt.

The order of the parameters does not matter. For example:

$*python3 align.py -i seq1.txt -j seq2.txt -o out.txt -p F -atype G*
and
$*python3 align.py -i seq1.txt -j seq2.txt -p F -atype G -o out.txt*

should give the same command.

The program should read two sequences and output the alignment of two sequences in the text file.

Your program should be able to parse the arguments correctly and can identify if there are

wrong or missing arguments, e.g., missing sequence file name after -i, etc.

For example:

seq1.txt contains the following information:

```
>seq1
ATGTTAT
```

seq2.txt contains the following information:

```
>seq2
ATCGTAC
```

Then the program will output the following information to out.txt (the exact alignment could be different depending on the scoring matrix used, the type of alignment, and also the way you chose the best score from):

```
seq1:   1 AT_GTTAT_  7
seq2:   1 ATCGT_A_C 7

Score: -15
Identities: 5/9 (56%)
```

The indexes give the start and end positions of such alignment. Since this is global alignment, they are just the first and last positions of both sequences.

Bonus problem (5 pts)

Change your program so that it can also perform local sequence alignment depending on the user's configuration.

- $*python3 align.py -i seq1.txt -j seq2.txt -o out.txt -p F -atype* ***L***
  Perform local pairwise sequence alignment of two DNA sequences in seq1.txt and seq2.txt. The output will be in out.txt.

If the sequences of the alignments are longer than 60 characters, then you may output 60 columns of the alignment per line. For example, the following gives an example of local alignment of two protein sequences (position 391-484 for protein P43609 and 932-1040 for protein P12270).

```
P43609: 391  ECVNDAVQTLLQGDDKLGKVSDKSREISEKYIEESQAIIQELVK-LTMEK---------L 440
P12270: 932  EDVDDLVSQLRQTEEQVNDLKERLK-TSTSNVEQYQAMVTSLEESLNKEKQVTEEVRKNI 990

P43609: 441  ESKFTKLCDLETQLEMEKLKYVKESEKMLNDRL----SLSKQILDLNKSL 486
P12270: 991  EVRLKESAEFQTQLEKKLMEVEKEKQELQDDKRRAIESMEQQLSELKKTL 1040

Score: 54
Identities:  27/110 (24%)
```

Again, the exact alignment could be different depending on the scoring matrix and the type of the alignment and the way you chose the best score from. For local alignments, sometimes you may find several alignments of subsequences of two sequences with same highest alignment scores. If that is the case, output all of these alignments.

Note:

1. For this program, you don't have to differentiate between gap opening penalty and gap extension penalty.
2. To test your program, you can use the provided sample files (i.e., dnaseq1.txt and dnaseq2.txt, proteinseq1.txt and proteinseq2.txt, and proteinseq3.txt and proteinseq4.txt) or download sequence files from the internet.
3. It might be helpful for you to perform alignments of short sequences by hand first (use the example sequences on slides/notes and provided scoring matrices), then compare with the alignments generated by the program to see if your program works correctly.
4. Submit all your files in a zip file. Provide a report which contains screenshots of running your program for various files.