**Background**. The purpose of this running exercise is to build python applications that align DNA sequences

**Goal**. We want to calculate how well two DNA sequences aligned to each other. If we have a match between two nucleotides in a given position, it will be scored +1. If we have a transition, a<->g or c<->t, the penalty will be -1, and if it is a transversion (any other substitution), the penalty will be -2. A gap penalty is -1. A figure showing transitions/transversions is found here: <https://en.wikipedia.org/wiki/Transversion#/media/File:Transitions-transversions-v4.svg>.

In this example input:

|  |
| --- |
| >id1  a c g t a g g t t t t a  >id2  a t g t - - g t t t - - |

The alignment score is 3:

|  |
| --- |
| a c g t a g g t t t t a  a t g t - - g t t t g a  **1** -1 **+1 +1** -1 -1 **+1** **+1** **+1** **+1** -2 **+1** = 3 |

Write a program that outputs the ids of the aligned sequences, their identity, gaps, and alignment score. If the fasta file contains multiple sequences, do an all-against-all calculation. For example, if the file has three sequences, print:

|  |
| --- |
| Id1-id2: Identity: 8/12 (66.7%), Gaps: 2/12 (16.7%), Score=3  Id1-id3: Identity: 4/12 (33.3%), Gaps: 2/12 (16.7%), Score=1  Id2-id3: Identity: 12/12 (100%), Gaps: 0/12 (0%), Score=12 |

The scores described above should be the default ones. Allow the user to provide a file parameters.txt with different scores.

**Data**. Data for the assignment can be downloaded from Canvas. The archive file *RunningExercise2-1.tgz* includes several examples of fasta files, but you should create more.

**Execution.** Your program should take one fasta file with two or more sequences and output a single file with the alignment scores for each pair. Your code should run like this:

|  |
| --- |
| python FastaAligner.py input\_fasta.fna parameters.txt [optional] output\_fasta.txt [optional] |

**Submission.**

Compress your code, input, and output files and call them *FastaAlignerProject-1.zip.*

Upload the compressed file to Canvas by the deadline.

**Import notes**.

1. Assume that the sequences are already aligned. You are not asked to write an alignment program, only to evaluate the alignment.
2. Your program should be completely robust for users who will try to crash it (us). Write in your documentation for all the cases that your code can handle and what decisions you made.