Aim: In this assignment, given two DNA sequences in a **single** FASTA-formatted file, we ask to implement both global and local alignment algorithms that use naïve and affine gap penalties. For all modes, you will use the following scoring matrix for match/mismatch score:

	A	C	G	T
A	2	-3	-3	-3
C	-3	2	-3	-3
G	-3	-3	2	-3
T	-3	-3	-3	2

Command line examples: Each line represents different runs for different modes.

```
allalign --mode global --input sequences.fasta --gapopen -5
allalign --mode aglobal --input sequences.fasta --gapopen -5 --gapext -2
allalign --mode local --input sequences.fasta --gapopen -5
allalign --mode alocal --input sequences.fasta --gapopen -5 --gapext -2
```

Input: All inputs will be used for all alignments.

- Same input file **"sequences.fasta"**, which contains two DNA sequences.

Parameters:

- −−**mode:** It will be selected from one of the followings:
  - **global:** Needleman-Wunsch with naïve gap scoring
  - **local:** Smith-Waterman with naïve gap scoring
  - **aglobal:** Needleman-Wunsch with affine gap scoring
  - **alocal:** Smith-Waterman with affine gap scoring
- −−**input:** Input FASTA file for sequences
- −−**gapopen:** Gap opening penalty for affine gap model, or unit gap cost for naïve model
- −−**gapext:** Gap extension penalty for affine gap model

Output:

- **global-naiveGap.aln** will be the only output file if the parameter is −−**mode global**
- **global-affineGap.aln** will be the only output file if the parameter is −−**mode aglobal**
- **local-naiveGap.aln** will be the only output file if the parameter is −−**mode local**
- **local-affineGap.aln** will be the only output file if the parameter is −−**mode alocal**

Input file format (sequences.fasta):

```
>my_first_sequence
TCGACCCAAGTAGGGAAAGAATATCAACACAAAGGCTCGAGAAGAGCCACC
CCATGAGCCACCGCATCTACCCCGTGCCCCAGCAAATTAAGAATAG
>another_sequence
TCGACCCATGTAGGGAAAGCATATCAATTTCACAAAGGCTCGAGAAGAGCC
ACATGAGCCACCGCATCTACCCCAGCAAATTAAGAAAAG
```

Output file format:

```
Score = 118

my_first_sequence       TCGACCCAAGTAGGGAAAGAATATCAA---CACAAAGGCTCGAGAAGAGCCACCCCATGA
another_sequence        TCGACCCATGTAGGGAAAGCATATCAATTTCACAAAGGCTCGAGAAGAGCCAC---ATGA

my_first_sequence       GCCACCGCATCTACCCCGTGCCCCAGCAAATTAAGAATAG
another_sequence        GCCACCGCATCTACCCC-------AGCAAATTAAGAAAAG
```

The example above is for **(−−mode aglobal) only**. We will **not** give examples for the other alignment options, but we require the alignments in the same format. If the alignment is long (>60 characters), partition the alignment into multiple lines. Each line should have at most 60 characters (not counting the sequence names). Scoring matrix will not be given as parameter so you can use it with **the same values** inside of your code.

Notes:

- You must write your code yourself. Sufficient evidence of plagiarism will be treated the same as for plagiarism or cheating.
- Non-compiling submissions will not be evaluated.
- Your code will be compiled into a **single binary** using the Makefile. If scripting languages are used, a single wrapper script should be provided.
- Do not submit the program binary. You must submit the following items:
  - All of the source files.
  - A script to compile the source code and produce the binary (Makefile), if required.
  - A README.txt file that describes how the compilation progress works, if required.
- Zip your files and send them in only one zipped file. File name format='surname_name_hw3.zip'
- **Submit your code through the Moodle page. DO NOT EMAIL.**
- C / C++, Python, Java will be used as programming language. STL is allowed. You may reuse the FASTA parsing code from your previous homeworks. Make sure your code compiles and works in Linux systems (gcc compiler for C/C++).
- All submissions must be made by 23:59, November 13, 2019. Late submissions will lose 20 points for each additional day. Three or more days of delay will result in a zero grade.