Aim: In this assignment, given **n** DNA sequences, $2 \leq n \leq 25$, in FASTA file, we ask to implement **Guide Tree Construction using the UPGMA algorithm**. You may utilize your own **Affine gap** implementation of **Needleman-Wunsch** algorithm from the third Homework or you can write it from the scratch. You will take gap opening and extension penalties, match score, and mismatch penalty via parameters. The length of the input sequences might be at most 500 characters. For the sake of simplicity, we give the alignment file (.fasta) for three sequences as an example described below. Output will be **in Newick tree format** (http://evolution.genetics.washington.edu/phylip/newicktree.html). Create your distance matrix by first aligning with Needleman-Wunsch, and then calculating edit distances of the pairwise alignments. Do **NOT** give the intermediate steps as output.

Command line examples: Be sure that your code works using the following command **(NOT PARSING the arguments will cost -15 points)**:

```
buildUPGMA --fasta sequences.fasta --match 5 --mismatch -3 --gapopen -8 --gapext -1 --out sequences.tree
```

Input:

- −−**fasta** FASTA-formatted file containing all sequences. This file may include up to 25 sequences. You may need to parse this file twice to count the number of sequences and then load them, if necessary.

```
>A
CTAGATAATTGCCAGATGATCAAATTTATAT
>B
CTAGATAATCATGCTAGCTAGTGCACAAATTTATAT
>C
CTAGATAATTGGAATGTCGATCGATCG
```

Parameters:

- −−**gapopen** gap opening penalty
- −−**gapext** gap extension penalty
- −−**match** match score
- −−**mismatch** mismatch penalty

Output:

- −−**out:** sequences.tree in Newick format.

```
((A:4.5, B:4.5):2.75, C:7.25);
```

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 **"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**.
- Create a directory, of which format is "surname_name_hw5", put all required files into that directory, and then zip it. You will have a give a single zipped file, 'surname_name_hw5.zip'.
- **Submit your code through the Moodle page. DO NOT EMAIL.**
- C / C++, Python, or Java will be used as programming language. STL is allowed. **You may reuse code(s) from your previous homework assignments.** Make sure your code compiles and works in Linux systems (gcc compiler for C/C++).
- All submissions must be made by 23:59, December 19, 2019. Late submissions will lose 20 points for each additional day. Three or more days of delay will result in a zero grade.