B2 - Synthesis Pool

B-SYN-200

FASTAtools

tools for DNA analysis

3.0

FASTAtools

binary name: requirement.c, FASTAtools
language: C
compilation: via Makefile, including re, clean and fclean rules

- The totality of your source files, except all useless files (binary, temp files, obj files,...), must be included in your delivery.
- All the bonus files (including a potential specific Makefile) should be in a directory named *bonus*.
- Error messages have to be written on the error output, and the program should then exit with the 84 error code (0 if there is no error).

Requirement

Write a C file named **requirement.c** containing a function that capitalizes the first letter of each word.
The function must be prototyped the following way:

```
char *my_strcapitalize_synthesis(char *str);
```

The function returns *str*.

The phrase, “hey, how are you? 42words forty-two; fifty+one” will become, “Hey, How Are You? 42words Forty-Two; Fifty+One”.

Only malloc and free are allowed from libC.
The rest of the project will not be corrected unless this requirement is fully functional (and rewritten).

The file must be placed at the root of your git repository.
It will be compiled with our main function, and our Makefile (the -I flag being empty).

FASTATOOLS

Bioinformatics is about creating softwares that cope with biology issues.
A classic issue consists in collecting genetic sequences (DeoxyriboNucleic Acid, DNA), and extracting as much information as possible, such as genes or coded proteins.
The goal of this project is to develop a library in order to compute basic DNA analyses.

DNA is a molecule found in the cells of living beings.
Parents transmit DNA to their progeny and, thanks to successive mutations, it is responsible for evolution; reading, decoding and understanding DNA leads to living beings understanding.

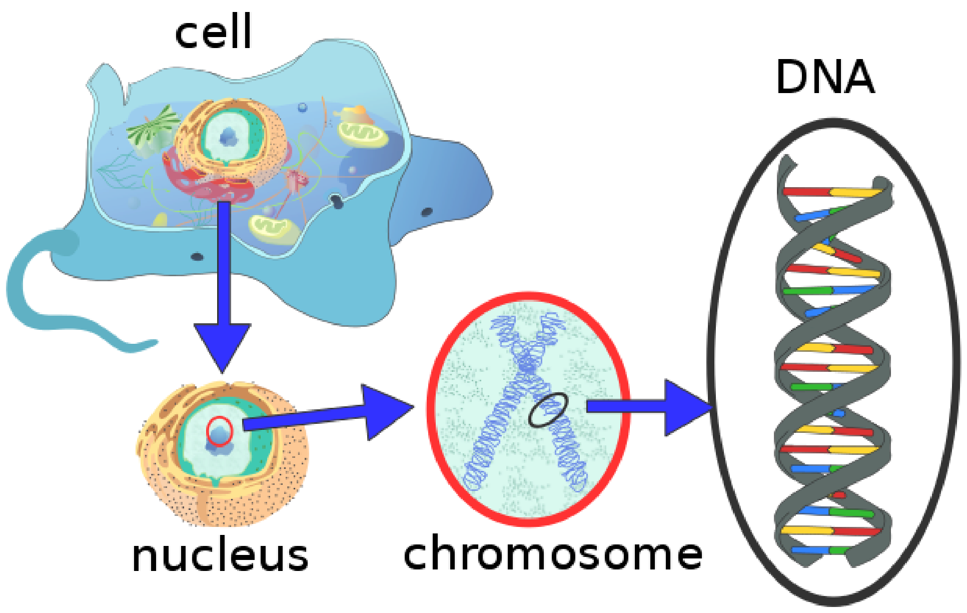


Figure 1: DNA

DNA is composed of 4 molecules: Adenine (A), Cytosine (C), Guanine (G) and Thymine (T).
Thus, DNA is often regarded as a string (called sequence), representing the sequency of those molecules.

THE FASTA FORMAT

FASTA is a file format often used in bioinformatics.
It consists in matching identifiers and DNA sequencies.
The identifier is declared on a line starting with the > symbol.
The following filled lines, which do not start with >, contain the DNA sequence.

```
Terminal
∼/B-SYN-200> cat example.fasta
>at1200347|szg01
atgatgctccctgatgtataatgccgcntctacttaaaatctcgatgctgccactgtgtctt
gctatgcggcggctacggggtattanccgatcaaatctgctgatcatctcgctccttnca

>at1102337|szg02
atGATGctccctgatgtgatgcagtNgagtacacttaAAATCTCGATGCtgccacgctgatcgtct
agctatgcggcggNtacggggtattaCCGATCAAgat
>at2430958|szg01

ATGATGCTCCCTGATGTATAATNCCGCTATCTACTTAAAATNTCGATGCTGCCACTGTGTCTTTCT
AGCTATGCGGCGNCTACGGGGTATTACCGATCAAATCTGCTGATCATCTCGNNCCTTGCTTCA
```

FASTA format is very tolerant! Mixed upper and lower cases, letters other than ATGC (N for instance, meaning sequence error), empty lines, free-form identifiers...

Read a FASTA file from the standard input, and write the DNA sequences to the standard output, while complying with the FASTA format.

```
Terminal
∼/B-SYN-200> ./FASTAtools -h
USAGE
      ./FASTAtools option

DESCRIPTION
      option 1: read FASTA from the standard input, write the DNA sequences to the
              standard output
```

From now on, only 'A', 'a', 'T', 't', 'G', 'g', 'C', 'c', 'N' and 'n' characters will be considered in the DNA sequences, and will be output capitalized.

```
Terminal
∼/B-SYN-200> cat ex.fasta
>seq1
ATATGCAGAGTTAGT
>seq2
catCATGCC TADGACGAA
GAGATGCTTTGAATGGRAATGAA
∼/B-SYN-200> ./FASTAtools 1 < ex.fasta
>seq1
ATATGCAGAGTTAGT
>seq2
CATCATGCCTAGACGAAGAGATGCTTTGAATGGAATGAA
```

RNA

RNA (RiboNucleic Acid) is a translation of DNA in which 'T's are replaced by 'U's.

Read a FASTA file from the standard input, and write the RNA sequences to the standard output while complying with the FASTA format.

```
Terminal
~/B-SYN-200> ./FASTAtools -h
USAGE
      ./FASTAtools option

DESCRIPTION
      option 1: read FASTA from the standard input, write the DNA sequences to the
              standard output
      option 2: read FASTA from the standard input, write the RNA sequences to the
              standard output
```

```
Terminal
~/B-SYN-200> cat ex.fasta
>seq1
ATATGCAGAGTTAGT
>seq2
catCATGCC TADGACGAA
GAGATGCTTTGAATGGRAATGAA
~/B-SYN-200> ./FASTAtools 2 < ex.fasta
>seq1
AUAUGCAGAGUUAGU
>seq2
CAUCAUGCCUAGACGAAGAGAUGCUUUGAAUGGAAUGAA
```

REVERSE COMPLEMENT

Actually, DNA is composed of two parallel strands that are helically wound.
Only one strand is coded in FASTA files, since 'T's and 'A's (as well as 'G's and 'C's)systematically come face to face.
Nevertheless, each strand is read in a different way.
The strand stocked in the FASTA file reads from left to right. The other one from right to left; this is called the **Reverse Complement**.

Read a FASTA file from the standard input, and write the reverse complement sequences to the standard output, while complying with the FASTA format.

Terminal

```
~/B-SYN-200> ./FASTAtools -h
USAGE
      ./FASTAtools option

DESCRIPTION
      option 1: read FASTA from the standard input, write the DNA sequences to the
              standard output
      option 2: read FASTA from the standard input, write the RNA sequences to the
              standard output
      option 3: read FASTA from the standard input, write the reverse complement
              to the standard output
```

Terminal

```
~/B-SYN-200> cat ex.fasta
>seq1
ATATGCAGAGTTAGT
>seq2
catCATGCC TADGACGAA
GAGATGCTTTGAATGGRAATGAA
~/B-SYN-200> ./FASTAtools 3 < ex.fasta
>seq1
ACTAACTCTGCATAT
>seq2
TTCATTCCATTCAAAGCATCTCTTCGTCTAGGCATGATG
```

K-MERS

A DNA sequence can be broken down into **k-mers**; that is to say, in substrings of length *k*.
For instance, the seqence `ATGTTGGCC` of length of 9 gives 7 3-mers : `ATG`, `TGT`, `GTT`, `TTG`, `TGG`, `GGC`, `GCC`.

Read a FASTA file from the standard input, and the integer k as an argument, and write the list of all k-mers (one per line) in the FASTA file and put in alphabetical order.

k-mers must be unique in the list!

Terminal

```
~/B-SYN-200> ./FASTAtools -h
USAGE
      ./FASTAtools option [k]

DESCRIPTION
      option 1: read FASTA from the standard input, write the DNA sequences to the
              standard output
      option 2: read FASTA from the standard input, write the RNA sequences to the
              standard output
      option 3: read FASTA from the standard input, write the reverse complement
              to the standard output
      option 4: read FASTA from the standard input, write the k-mer list to the
              standard output
      k: size of the k-mers for option 4
```

Terminal

```
∼/B-SYN-200> cat ex.fasta
>seq1
ATATGCAGAGTTAGT
>seq2
catCATGCC TADGACGAA
GAGATGCTTTGAATGGRAATGAA
∼/B-SYN-200> ./FASTAtools 4 1 < ex.fasta
A
C
G
T
∼/B-SYN-200> ./FASTAtools 4 29 < ex.fasta
AGACGAAGAGATGCTTTGAATGGAATGAA
ATCATGCCTAGACGAAGAGATGCTTTGAA
ATGCCTAGACGAAGAGATGCTTTGAATGG
CATCATGCCTAGACGAAGAGATGCTTTGA
CATGCCTAGACGAAGAGATGCTTTGAATG
CCTAGACGAAGAGATGCTTTGAATGGAAT
CTAGACGAAGAGATGCTTTGAATGGAATG
GCCTAGACGAAGAGATGCTTTGAATGGAA
TAGACGAAGAGATGCTTTGAATGGAATGA
TCATGCCTAGACGAAGAGATGCTTTGAAT
TGCCTAGACGAAGAGATGCTTTGAATGGA
```

Coding sequence

A **codon** is a series of 3 adjacent DNA bases.
There are 3 different types of codons:

- ATG is the start codon, whose role is to start a coding sequence,
- TAA, TAG and TGA are the stop codons, whose role is to end a coding sequence,
- all the other possible codons, representing 20 amino acids (since different codons can represent the same amino acid).

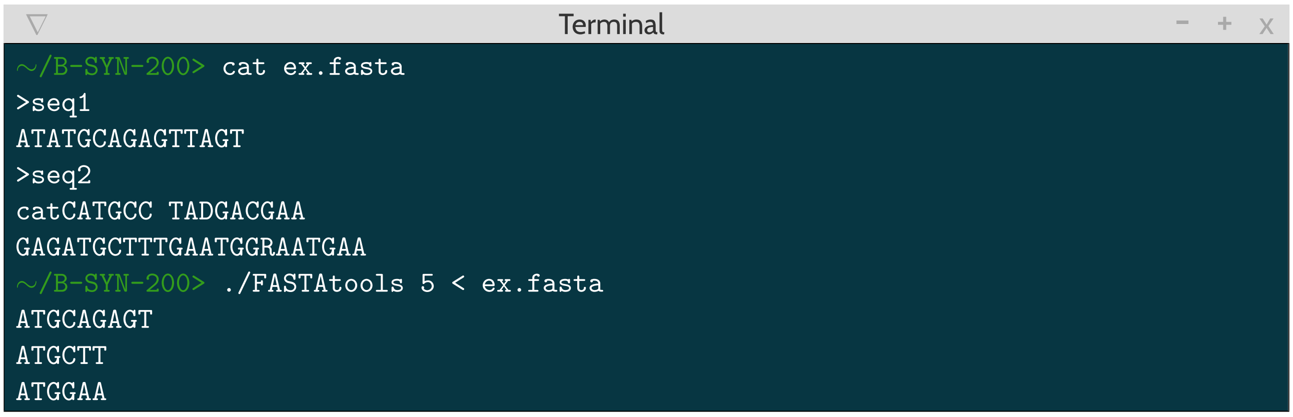
A coding sequence is thus created by the start codon, then the following amino acids, until the stop codon (which is not part of the coding sequence).
There can be several coding sequences in a DNA sequence.
For instance, the sequence `TGTCATGGCGTGCGATAGACGTCATTAGTTA` gives (the start and stop codons being bolded):
`...T GTC **ATG** GCG TGC GAT AGA CGT CAT **TAG** TTA.`
Thus, the coding sequence is: `ATGGCGTGCGATAGACGTCAT`.

Read a FASTA file from the standard input, and write the list of all coding sequences (one per line) in the FASTA file, put in alphabetical order.

Coding sequences can be found on one, or another, strand!

```
Terminal
∼/B-SYN-200> ./FASTAtools -h
USAGE
      ./FASTAtools option [k]

DESCRIPTION
      option 1: read FASTA from the standard input, write the DNA sequences to the
              standard output
      option 2: read FASTA from the standard input, write the RNA sequences to the
              standard output
      option 3: read FASTA from the standard input, write the reverse complement
              to the standard output
      option 4: read FASTA from the standard input, write the k-mer list to the
              standard output
      option 5: read FASTA from the standard input, write the coding sequences
              list to the standard output
      k: size of the k-mers for option 4
```



```
~/B-SYN-200> cat ex.fasta
>seq1
ATATGCAGAGTTAGT
>seq2
catCATGCC TADGACGAA
GAGATGCTTTGAATGGRAATGAA
~/B-SYN-200> ./FASTAtools 5 < ex.fasta
ATGCAGAGT
ATGCTT
ATGGAA
```

AMINO ACIDS

Each coding sequence represents a series of **amino acids** (which form proteins).

Using the corresponding table below, read a FASTA file from the standard input, and write the alphabetically-ordered list of all amino acids (one per line) in the FASTA file.

amino acid	**codon(s)**
A	GCT, GCC, GCA, GCG
C	TGT, TGC
D	GAT, GAC
E	GAA, GAG
F	TTT, TTC
G	GGT, GGC, GGA, GGG
H	CAT, CAC
I	ATT, ATC, ATA
K	AAA, AAG
L	TTA, TTG, CTT, CTC, CTA, CTG
M	ATG
N	AAT, AAC
P	CCT, CCC, CCA, CCG
Q	CAA, CAG
R	AGA, AGG, CGT, CGC, CGA, CGG
S	TCT, TCC, TCA, TCG, AGT, AGC
T	ACT, ACC, ACA, ACG
V	GTT, GTC, GTA, GTG
W	TGG
X	any codon containing N
Y	TAT, TAC

```
~/B-SYN-200> ./FASTAtools -h
USAGE
      ./FASTAtools option [k]

DESCRIPTION
      option 1: read FASTA from the standard input, write the DNA sequences to the
              standard output
      option 2: read FASTA from the standard input, write the RNA sequences to the
              standard output
      option 3: read FASTA from the standard input, write the reverse complement
              to the standard output
      option 4: read FASTA from the standard input, write the k-mer list to the
              standard output
      option 5: read FASTA from the standard input, write the coding sequences
              list to the standard output
      option 6: read FASTA from the standard input, write the amino acids list to
              the standard output
      k: size of the k-mers for option 4
```

```
~/B-SYN-200> cat ex.fasta
>seq1
ATATGCAGAGTTAGT
>seq2
catCATGCC TADGACGAA
GAGATGCTTTGAATGGRAATGAA
~/B-SYN-200> ./FASTAtools 6 < ex.fasta
ME
ML
MQS
```

DNA sequences alignment

In order to find out how similar two organisms are, their DNA sequences need to be compared. It is necessary to align them.
Several algorithms can be used to perform this alignment, such as the Needleman-Wunsch algorithm.

Add the possibility to align DNA sequences (using the mentioned algorithm) to your software.
Display the alignement score using the following parameters : `match`=1, `mismatch`=-1, `gap`=-1.

Terminal

```
~/B-SYN-200> ./FASTAtools -h
USAGE
      ./FASTAtools option [k]

DESCRIPTION
      option 1: read FASTA from the standard input, write the DNA sequences to the
              standard output
      option 2: read FASTA from the standard input, write the RNA sequences to the
              standard output
      option 3: read FASTA from the standard input, write the reverse complement
              to the standard output
      option 4: read FASTA from the standard input, write the k-mer list to the
              standard output
      option 5: read FASTA from the standard input, write the coding sequences
              list to the standard output
      option 6: read FASTA from the standard input, write the amino acids list to
              the standard output
      option 7: read FASTA from the standard input containing exactly 2 squences,
              align them and write the result to the standard output
      k: size of the k-mers for option 4
```