1. Assignment 1 (7.5 marks total)

Prolog is probably unlike any other programming language you have used until now. You should try to develop prolog code as logic definitions. This will be frustrating at first, but when you finally get the hang of it, you will find that you can write much more understandable code.

Prolog supports rapid prototyping. While Prolog programs can be made to be efficient, it often decreases the understandability of the program. You should start with a program that is "obviously correct" and then you can transform it into a more efficient program as needed.

The logic needed for the first two questions was introduced in the 1st week, the prolog needed for the programming question will be introduced in the 2nd week, and the assignment is submitted on August the 21st. You will not have a lot of time for the programming part of the assignment, so you must plan your time accordingly.

1.1 Propositional Logic Problem (2 marks)

DNA is made up of the four nucleotide bases Adenine (A), Thymine (T), Cytosine (C) and Guanine (G). Adenine and Guanine are purines. Cytosine and thymine are pyrimidines. Pyrimidines always pair with purines. Express these constraints in propositional logic

- If A is a base, it is either a purine or a pyrimidine, but not both; similarly for T C and G. Example:
  - base_A ⇒ purine_A ∨ pyrimidine_A ∧ ¬ (purine_A ∧ pyrimidine_A)
- If A bonds with C (A_bondswith_C), it does not bond with T or G. Similarly for the other bases
- A does not bond with A; similarly C, T or G do not bond with themselves
- If A bonds with T, it is the case that A is a base and that T is a base, and it is either the case that A is a purine and T a pyrimidine, or vice versa. Similarly with other pairs of bases.
- If A bonds with T, then T bonds with A, and similarly for all other pairs

An important feature of formulas in propositional logic is whether they are satisfiable or not. That is, whether there is an assignment of the truth values {true, false} to the propositions (e.g. A_bondswith_C) such that the entire formula evaluates to true. Choose such an assignment to the propositions in your formulas, and show that they each evaluate to true using a truth table. It might be helpful to use the facts of biology as a guide to choosing which propositions are *true* and which are *false*; basic information about the biology of DNA and RNA is widely available on the Internet. For example, your predicates for “A bonds with T” and for “G bonds to C” should be *true.*

1.2. Predicate Logic Problem (1 mark)

Write a general version of the bonds-with relation, bondsWith(Base1,Base2), in predicate calculus only using standard quantifiers, connectives, the unary not (¬), equality (=) and the predicates:

- base(X)
- pyrimidine(X); purine(X)

And the constants:

- adenine, thymine, cytosine and guanine

1.3. Prolog Logic Problem (4.5 Marks 367)

Translate your predicate calculus definition of bondsWith into a Prolog program. As in your predicate definitions, only use the predicates mentioned above. Test your program by asserting that bondsWith(adenine,thymine) and showing that bondsWith(cytosine, WHAT) returns that WHAT is guanine.

Now extend your program:
This paper: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7151553/ (which will also be made available on canvas) describes a protein amino acid subsequence KRSFIEDLLFNKV of the S2′ spike glycoprotein proteolytic cleavage site which appear to be important to COVID-19's infection strategy.

The data at https://www.ncbi.nlm.nih.gov/nuccore/MT072688 gives the RNA genome using DNA coding for a particular variant of the COVID-19 virus, starting at this part of the file:

```
ORIGIN
        1 cttcccaggt aacaaaccaa ccaactttcg atctcttgta gatctgttct ctaaacgaac
       61 tttaaaatct gtgtggctgt cactcggctg catgcttagt gcactcacgc agtataatta
      121 ataactaatt actgtcgttg acaggacacg agtaactcgt ctatcttctg caggctgctt
      181 acggtttcgt ccgtgttgca gccgatcatc agcacatcta ggtttcgtcc gggtgtgacc
      241 gaaaggtaag atggagagcc ttgtccctgg tttcaacgag aaaacacacg tccaactcag
      301 tttgcctgtt ttacaggttc gcgacgtgct cgtacgtggc tttggagact ccgtggagga
      361 ggtcttatca gaggcacgtc aacatcttaa agatggcact tgtggcttag tagaagttga
```

https://en.wikipedia.org/wiki/DNA_codon_table gives a table, reproduced below, mapping this code onto the amino acids that make up proteins. For example the "atg" that starts at position 251 the above sequence codes for Methionine (or M) [the section before that, highlighted in the excerpt, is called the 5' untranslated region, and is a kind of file header].

Your task: Define predicates and rules in Prolog that encode the coding table given below, and, using that information, tests whether a particular protein amino acid sequence is coded for in a

particular viral genome. You should support a predicate codesFor/2 that returns true if the DNA sequence referred to in its first argument (e.g. mt072688) codes for the protein amino acid sequence in its second argument (e.g: codesFor(mt072688,"KRSFIEDLLFNKV"). The closer your code comes to working with the FASTA file for the genome (e.g. https://www.ncbi.nlm.nih.gov/nuccore/MT072688.1?report=fasta) the better.
You may find it useful to use the SWI-Prolog built-in predicate string_chars/2 documented at https://www.swi-prolog.org/pldoc/doc_for?object=string_chars/2

Amino-acid biochemical properties	Nonpolar	Polar	Basic	Acidic

Termination: stop codon

Standard genetic code

1st base	2nd base								3rd base
	T		C		A		G		
T	TTT	(Phe/F) Phenylalanine	TCT	(Ser/S) Serine	TAT	(Tyr/Y) Tyrosine	TGT	(Cys/C) Cysteine	T
	TTC		TCC		TAC		TGC		C
	TTA	(Leu/L) Leucine	TCA		TAA	Stop (*Ochre*)[B]	TGA	Stop (*Opal*)[B]	A
	TTG[A]		TCG		TAG	Stop (*Amber*)[B]	TGG	(Trp/W) Tryptophan	G
C	CTT		CCT	(Pro/P) Proline	CAT	(His/H) Histidine	CGT	(Arg/R) Arginine	T
	CTC		CCC		CAC		CGC		C
	CTA		CCA		CAA	(Gln/Q) Glutamine	CGA		A
	CTG[A]		CCG		CAG		CGG		G
A	ATT	(Ile/I) Isoleucine	ACT	(Thr/T) Threonine	AAT	(Asn/N) Asparagine	AGT	(Ser/S) Serine	T
	ATC		ACC		AAC		AGC		C
	ATA		ACA		AAA	(Lys/K) Lysine	AGA	(Arg/R) Arginine	A
	ATG[A]	(Met/M) Methionine	ACG		AAG		AGG		G
G	GTT	(Val/V) Valine	GCT	(Ala/A) Alanine	GAT	(Asp/D) Aspartic acid	GGT	(Gly/G) Glycine	T
	GTC		GCC		GAC		GGC		C
	GTA		GCA		GAA	(Glu/E) Glutamic acid	GGA		A
	GTG		GCG		GAG		GGG		G

2. Submission Information

1. What to submit

You need to submit a zip archive, yourUpi.zip (e.g., mbar098.zip) containing 2 files: LogicProblems.pdf, and spikeprotein.pl. The first file has the answers for questions 1 and 2. The second one is the genetic analysis prolog code.

2. When and where to submit

You need to submit this to Canvas by 21 August 23:59. You will be informed when Canvas is ready to receive submissions.

3. Marking Rubric

1. Written answer marking

The marking for the propositional and predicate logic modelling questions is straightforward.

2. Protein sequence Prolog question

Your prolog program for this question MUST figure out whether the FASTA genome codes for the protein variant FOR ITSELF!!!!! You will get a zero for this question, if the answer is hard-coded into the prolog code. You also must translate the translation tables, etc., into prolog code yourself, see warning below.

3. WARNING

You may be tempted to just use someone else's code, DON'T!!!! The assignment you submit MUST BE your own work!! You can talk about it with others and we recommend that you do, but if we detect that you copied any of the assignment from another source, you will get a ZERO for the ENTIRE ASSIGNMENT!! This should not come as news for you; it just recapitulates basic standards of academic integrity, with which you will be thoroughly familiar by now.