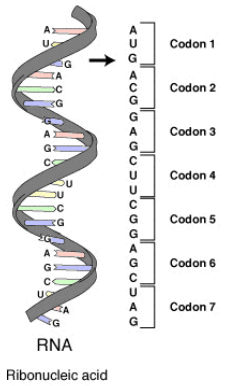
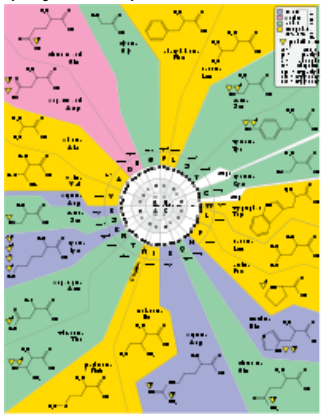
Genetic code

The **genetic code** consists of a number of lines that determine how living cells translate the information coded in genetic material (DNA or RNA sequences) to proteins (amino acid sequences). This code defines how a sequence of three nucleotides — named **codons** — specifies which amino acid will be added next to the protein during the protein synthesis.



A codon sequence within a messenger RNA (mRNA) molecule. Every codon consists of three nucleotides, that generally represent only one amino acid. The nucleotides are abbreviated with the letters A, U, G, and C. This is mRNA, that uses U (uracil), as opposed to DNA, which uses T (thymine). This mRNA molecule will instruct to synthesize a protein according to this code.



The standard genetic code.

Because the majority of the genes use the same code, this specific code is often referred to as the canonical or standard code, or simply the genetic code, while various variant of the code have developed. The protein synthesis in human mitochondria is an example where a genetic code is used that deviate from the standard genetic code.

Living cells use 20 types of amino acids to code proteins, that each are appointed their own uppercase letter. With four different nucleotides, a code that starts with 2 nucleotides can code a maximum of 4^2 or 16 different amino acids. Genetic codes are 3-letter codes where some codons are portrayed on the same amino acids or are used as stop codon. A certain genetic code can be (partially) recorded by linking every one of the 64 possible nucleotides to an amino acid (indicated by an uppercase letter) or a stop codon (indicated with an asterisk (*)).

Assignment

Define a class GeneticCode with the following methods:

- An initializing method __init__ to which the location of a text file must be given as an argument. This text file must contain the translation table of the genetic code and must consist of 64 text lines. On the first line are a codon and the corresponding amino acid, separated by a space. You may assume that this file contains all 64 codons. However, the DNA (with T for thymine) or the RNA alphabet (with U for uracil) can be used for the codons. The example below uses a file in which the DNA alphabet is used. The newly made object must keep the information from the given translation table.
- A method aminoacid to which a valid codon (DNA or RNA) must be given as a string argument. The method must print an uppercase letter that represents the corresponding amino acid in the genetic code given. In this translation, the method may not make a distinction between uppercase and lowercase letters for the codon given, also, there may not be a distinction between the letters U (uracil) and T (thymine). This way, both DNA and RNA codons can be given to the method. Look at the example below to verify how the method should react if the given argument doesn't represent a valid codon.
- A method protein to which a string argument must be given that represents a DNA or RNA sequence. This string may contains both uppercase and lowercase letters, furthermore it contains only letters from the DNA or RNA alphabet. Watch the example to verify how the method should react if the given argument doesn't represent a valid DNA or RNA sequence. If a valid argument is given, the function must print the translation of the given DNA or RNA sequence to the corresponding protein sequence. The string that is given as a result must consist of only uppercase letters. If the length of the given DNA or RNA sequence is not a multiple of three, the last or the two last letters of the string must be ignored in the translation.

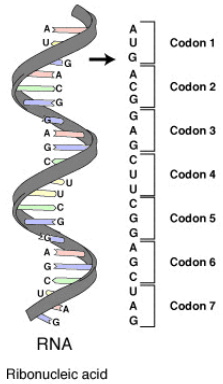
Example

In the example below, we assume that the file standard_code.txt is situated in the current directory.

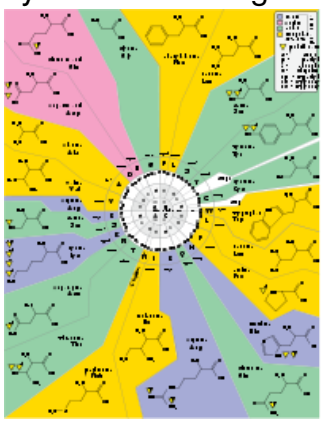
```
>>> code = GeneticCode('standard_code.txt')

>>> code.aminoacid('AGT')
'S'
>>> code.aminoacid('cga')
'R'
>>> code.aminoacid('UCU')
'S'
>>> code.aminoacid('ABC')
Traceback (most recent call last):
AssertionError: 'ABC' is not a valid codon.
>>> code.aminoacid('aagc')
Traceback (most recent call last):
AssertionError: 'aagc' is not a valid codon.

>>> code.protein('ATGCTGATGATGGGCTATTATCGAT')
'MLMMGYYR'
>>> code.protein('uauccuaguguc')
'YPSV'
>>> code.protein('AAGTCGTAGCTACGXXXXGAGAAGGAT')
Traceback (most recent call last):
AssertionError: invalid DNA or RNA sequence.
```

De **genetische code** bestaat uit een aantal regels die vastleggen hoe levende cellen de informatie gecodeerd in genetisch materiaal (DNA- of RNA-sequenties) vertalen naar eiwitten

(aminozuursequenties). Deze code definieert hoe een reeks van drie nucleotiden — **codons** genaamd — tijdens de eiwitsynthese specificeert welk aminozuur als volgende aan het eiwit zal toegevoegd worden.



Een codonreeks binnen een boodschapper RNA (mRNA) molecule. Elk codon bestaat uit drie nucleotiden, die doorgaans één enkel aminozuur voorstellen. De nucleotiden worden afgekort met de letters A, U, G, en C. Dit is mRNA, dat U (uracil) gebruikt. DNA maakt in plaats daarvan gebruik van T (thymine). Dit mRNA-molecuul zal een ribosoom instrueren om een eiwit te synthetiseren volgens deze code.


De standaard genetische code.

Omdat de overgrote meerderheid van de genen exact dezelfde code gebruiken, wordt vaak naar deze specifieke code gerefereerd als de canonische of standaard genetische code, of zelfs eenvoudigweg als dé genetische code, terwijl er zich in feite verschillende variante codes ontwikkeld hebben. De eiwitsynthese in menselijke mitochondria is een voorbeeld waar een genetische code gebruikt wordt die afwijkt van de standaard genetische code.

Levende cellen gebruiken 20 verschillende aminozuren om eiwitten te coderen, die elk een eigen hoofdletter toegewezen krijgen. Met vier verschillende nucleotiden kan een code die vertrekt van 2 nucleotiden slechts maximaal 4^2 of 16 verschillende aminozuren coderen. Een code van 3 nucleotiden kan daarentegen maximaal 4^3 of 64 verschillende aminozuren coderen. Genetische codes zijn dan ook 3-letter codes waarbij sommige codons op hetzelfde aminozuur afgebeeld worden, of fungeren als een stopcodon. Een bepaalde genetische code kan dus (deels) vastgelegd worden door elk van de 64 mogelijke nucleotiden te koppelen aan een aminozuur (aangegeven door een hoofdletter) of een stopcodon (aangeduid door een sterretje (*)).

Opgave

Definieer een klasse GenetischeCode met volgende methoden:

- Een initialisatiemethode `__init__` waaraan de locatie van een tekstbestand als argument moet doorgegeven worden. Dit tekstbestand moet de vertaaltabel van een genetische code bevatten en moet bestaan uit 64 tekstregels. Op elke regel staan een codon en het corresponderende aminozuur, van elkaar gescheiden door een spatie. Je mag ervan uitgaan dat het bestand alle 64 codons bevat. Voor de codons kan echter gebruik gemaakt worden van het DNA- (met `T` voor thymine) of het RNA-alfabet (met `U` voor uracil). Onderstaand voorbeeld gebruikt een bestand waarin het DNA-alfabet gebruikt wordt. Het nieuw aangemaakte object moet de informatie uit de gegeven vertaaltabel bijhouden.
- Een methode `aminozuur` waaraan een geldig codon (DNA of RNA) als stringargument moet doorgegeven worden. De methode moet een hoofdletter teruggeven die staat voor het corresponderende aminozuur in de gegeven genetische code. Bij deze vertaling mag de methode voor het gegeven codon geen onderscheid maken tussen hoofdletters en kleine letters, en ook niet tussen de letters `U` (uracil) en `T` (thymine). Op die manier kunnen zowel DNA- als RNA-codons aan de methode doorgegeven worden. Bekijk onderstaand voorbeeld om na te gaan hoe de methode moet reageren indien het doorgegeven argument geen geldig codon voorstelt.
- Een methode `eiwit` waaraan een stringargument moet doorgegeven worden dat een DNA- of RNA-sequentie voorstelt. Deze string mag zowel hoofdletters als kleine letters bevatten, maar bestaat voorts enkel uit letters uit het DNA- of RNA-alfabet. Bekijk onderstaand voorbeeld om na te gaan hoe de methode moet reageren indien het doorgegeven argument geen geldige DNA- of RNA-sequentie voorstelt. Wanneer een geldig argument wordt doorgegeven, moet de functie de vertaling van de gegeven DNA- of RNA-sequentie naar de corresponderende eiwitsequentie teruggeven. De string die als resultaat wordt teruggegeven mag hierbij enkel uit hoofdletters bestaan. Indien de lengte van de gegeven DNA- of RNA-sequentie geen veelvoud is van drie, dan moeten de laatste of de laatste twee letters van de string genegeerd worden bij de vertaling.

Voorbeeld

Bij onderstaand voorbeeld gaan we ervan uit dat het bestand standaard_code.txt zich in de huidige directory bevindt.

```
>>> code = GenetischeCode('standaard_code.txt')

>>> code.aminozuur('AGT')
'S'
>>> code.aminozuur('cga')
'R'
>>> code.aminozuur('UCU')
'S'
>>> code.aminozuur('ABC')
Traceback (most recent call last):
AssertionError: 'ABC' is geen geldig codon.
>>> code.aminozuur('aagc')
Traceback (most recent call last):
AssertionError: 'aagc' is geen geldig codon.

>>> code.eiwit('ATGCTGATGATGGGCTATTATCGAT')
'MLMMGYYR'
>>> code.eiwit('uauccuaguguc')
'YPSV'
>>> code.eiwit('AAGTCGTAGCTACGXXXXGAGAAGGAT')
Traceback (most recent call last):
```

AssertionError: ongeldige DNA- of RNA-sequentie.