Bioinformatyka - Laboratorium nr.2

Zadanie 1 - Łączenie i konwersja zmiennych.

a) Sprawdź wynik komend:

» print('The answer is %s'%42) lub użyj » print('The answer is', str(42))

b) wykonaj konwersję dowolnych 3-różnych zmiennych, korzystając z 3-ch dowolnych funkcji: int(x) - konwersja zmiennej x do int, long(x) - konwersja zmiennej x do long-a, float(x) - konwersja x do float'a, complex(real [,imag]) - zapis liczby zespolonej, str(x) - konwersja do string zmiennej x, tuple(s) - konwersja do krotki, list(s) - konwersja do listy, chr(x) - konwersja integer do znaku, unichr(x)- konwersja integer do Unicode, hex(x)-konwertuje integer do postaci hexadecymalnej, oct(x)-konwertuje integer do postaci ósemkowej

Zadanie 2 Słowniki.

a) W skład białek wchodzi 20 różnych rodzajów aminokwasów białkowych, każdy z nich ma skrót 1 lub 3 literowy np. A-Ala-alanina (patrz Rys.1). Poniżej przedstawiono kilka z nich w zbiorze o nazwie *amino*. Sprawdź wynik komend:

» amino = {'A':'Ala','C':'Cys','E':'Glu'} lub użyj »amino = dict(A='Ala', C='Cys', D='Glu')

» print(" C jest oznaczeniem dla: "+amino['C'])

»amino['A'] »amino.items() »amino.keys() »amino.values()

oraz

Przykład

for aa **in** amino.values():

print(aa)

A	Ala	alanina	M	Met	metionina
C	Cys	cysteina	N	Asn	asparagina
D	Asp	kw. asparaginowy	P	Pro	prolina
E	Glu	kw. glutaminowy	Q	Gln	glutamina
F	Phe	fenyloalanina	R	Arg	arginina
G	Gly	glicyna	S	Ser	seryna
H	His	histydyna	T	Thr	treonina
I	Ile	izoleucyna	V	Val	walina
K	Lys	lizyna	W	Trp	tryptofan
L	Leu	leucyn	Y	Tyr	tyrozyna

Rysunek 1: Aminokwasy

b) określ wynik działań następujących operatorów tj.$\&, |, -, \wedge$ na poniższe zbiory:

»zbior1 =set(['CP0.1','EF3.1','EF3.1'])

»zbior2 = set(['CP0.2','EF3.1','EF3.2'])

»wynik = zbior1 & zbior2 itd.

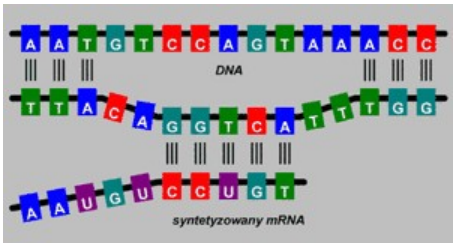
Pytanie: jakim operacjom matematycznym na zbiorach odpowiadają rezultaty działań w/w funkcji.

Zadanie 3 Moduły

Utwórz funkcję o nazwie **silnia**, która obliczy wartość silni dla zmiennej zdeklarowanej przez użytkownika, zapisz w/w funkcję do pliku *sil.py*, następnie napisz skrypt który obliczy dwumian Newtona n!/(k!(n-k)!) wykorzystując moduł *sil*. Uwaga: pamiętaj o **import** *nazwa modułu*.

Zadanie 4 Sekwencje DNA

Sekwencja DNA jest to kolejność nukleotydów w cząsteczce DNA. Oznaczana jest za pomocą skrótów od zasad wchodzących w skład nukleotydów. Analogicznie, sekwencja RNA jest to kolejność nukleotydów w cząsteczce RNA (w RNA nie występuje tak jak w DNA tymina "T", zamiast niej występuje uracyl "U"). Przykład: gdyby nić kodująca DNA była określona jako CCGATTACGT (A-adeina, G-guanina, C-cytozyna, T-tymina) to powyższa sekwencja mRNA wyglądałaby następująco: CCGAUUACGU.



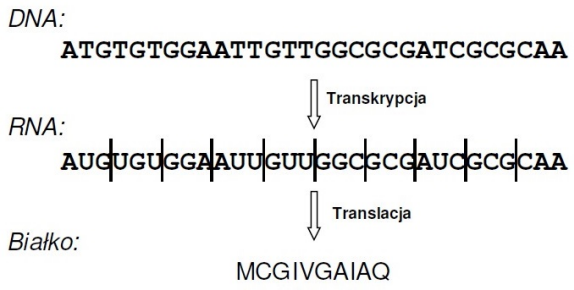
Rysunek 2: Proces transkrypcji, czyli przepisywanie informacji zawartej w DNA na RNA

a) zapoznaj się z ilustracją Rys.2 przedstawiającą proces transkrypcji (górną nić DNA nazywamy matrycową), zwróć uwagę na połączenia nici DNA.

Twoim zadaniem jest utworzenie programu, który umożliwi wykonanie użytkownikowi transkrypcji kodu DNA na RNA dla podanej przez użytkownikownika sekwencji DNA (nić matrycowa).
b) zmodyfikuj wcześniej napisany program tj. poinformuj użytkownika o niepoprawności danych jeśli poda liczbę nukleotydów mniejszą niż 10 lub większą niż 25.

Zadanie 5 Aminokwasy

Aminokwasy są budulcem białek. Każdy z aminokwasów ma indywidualną wagę. Podczas tworzenia białka, każde wiązanie aminokwasu uwalnia cząsteczki wody (o masie cząsteczkowej M=18). Aby obliczyć masę białka należy odjąć masę wszystkich uwalnianych cząsteczek wody od całkowitej masy aminokwasów wchodzących w skład białka.
a) wczytaj plik tekstowy zawierający masę poszczególnych aminokwasów **aminowaga.txt** zawierający dane:
{"A":89,"V":117,"L":131,"I":131,"P":115,"F":165,
"W":204,"M":149,"G":75,"S":105,"C":121,"T":119,
"Y":181,"N":132,"Q":146,"D":133,"E":147,
"K":146,"R":174,"H":155}
Twoim zadaniem jest utworzenie programu, który umożliwi obliczenie masy białka dla podanej przez użytkownika sekwencji aminokwasów. Przykładowa sekwencja aminokwasów np. MKTFVLHIFIFALVAF
b) dla zbioru=UUCUCGGACCUGGAGAUUCACAGU, oblicz ile razy w w/w zbiorze występują kombinacje liter: UCU, UCC, UCA, UCG, AGU lub AGC
c) Każdy aminokwas jest zbudowany z kodonów, każdy kodon składa się z 3 nukleotydów (patrz Rys.3), aminokwas o nazwie seryna składa się z kodonu: UCU, UCC, UCA, UCG, AGU lub AGC. Znając fragment sekwencji nukleotydów białka tj. UUCUCGGACCUGGAGAUUCACAGU napisz program który ustali ile razy występuje w nim aminokwas seryna, rozważ następujące przypadki podziału:
- UUC UCG GAC CUG GAG AUU CAC AGU
- U UCU CGG ACC UGG AGA UUC ACA GU
- UU CUC GGA CCU GGA GAU UCA CAG U



Rysunek 3: Proces translacji

PRZYPOMINAMY FUNKCJE

TABLE 3.2: Methods Associated with Dictionaries

Properties	Description
len(a)	Number of elements of *a*
a[k]	The element from *a* that has a *k* key
a[k] = v	Set a[k] to *v*
del a[k]	Remove a[k] from *a*
a.clear()	Remove all items from *a*
a.copy()	A copy of *a*
k in a	True if a has a key *k*, else False
k not in a	Equivalent to not *k* in *a*
a.has_key(k)	Equivalent to *k* in *a*, use that form in new code
a.items()	A copy of *a*'s list of (key, value) pairs
a.keys()	A copy of *a*'s list of keys
a.update([b])	Updates (and overwrites) key/value pairs from *b*
a.fromkeys(seq[, value])	Creates a new dictionary with keys from *seq* and values set to *value*
a.values()	A copy of *a*'s list of values
a.get(k[, x])	a[k] if *k* in *a*, else *x*
a.setdefault(k[, x])	a[k] if *k* in *a*, else *x* (also setting it)
a.pop(k[, x])	a[k] if *k* in *a*, else *x* (and remove *k*)
a.popitem()	Remove and return an arbitrary (key, value) pair

TABLE 3.1: Common List Operations

Properties	Description
s.append(x)	Adds the *x* element to list *s*
s.count(x)	Counts how many times *x* is in *s*
s.index(x)	Returns where is *x* in list *s*
s.remove(x)	Removes the element *x* from list *s*
s.reverse()	Reverse list *s*
s.sort()	Sort list *s*

Rysunek 4: