

1 Projekty zaliczeniowe (wersja robocza)

1.0 masa molowa

Napisz moduł `molarmass` z funkcją `mm()` która wylicza masę molową podanego związku. Funkcja powinna działać dla pierwiastków od H do Bi, tak żeby dało się jej użyć jak poniżej:

```
from molarmass import mm
print( mm('C2 H5 O H') )
print( mm('C8H10N4O2') )
print( mm('Na0.75K0.25Cl1I0.002') )
```

1.1 ile wody płynie w Wiśle

Pod adresem https://dane.imgw.pl/data/dane_pomiarowo_obserwacyjne/dane_hydrologiczne/dobowe/ są dane z wodowskazów w całej Polsce z ostatnich 50 lat. Napisz program, który rysuje poziom wody w Wiśle (wodowskaz Kraków-Bielany) w latach hydrologicznych 2016–2018. Kiedy były powodzie? Potrzebne będzie: ręczne ściągnięcie 36 plików zip, rozpakowanie jednego i rozpoznanie jakie dane w formacie csv są tam zapisywane. Program powinien w locie rozpakowywać i wczytywać niektóre dane z tych plików, i robić odpowiedni wykres. Ponadto program powinien wyliczyć ile wody przepłynęło przez Kraków w roku 2017.

1.2 genom wirusa SARS-CoV-2

Genom wirusa SARS-CoV-2 został opublikowany i jest dostępny w bazie <https://www.ncbi.nlm.nih.gov/nucleotide/MN908947> Więcej szczegółów w artykule z 19 lutego 2020 *Coronavirus and the race to distribute reliable diagnostics* <https://www.nature.com/articles/d41587-020-00002-2> Inny artykuł <https://academic.oup.com/clinchem/advance-article/doi/10.1093/clinchem/hvaa029/5719336> podaje krótkie sekwencje genów których można użyć do diagnostyki metodą PCR. Są to: *primers (forward and reverse)* oraz *probes*. Znajdź te sekwencje w genomie SARS-CoV-2. Może to wymagać odpowiednich modyfikacji tych sekwencji. Znajdź czy te same sekwencje są w genomie wirusa SARS z 2003 roku.

1.3 częstość liter

Napisz funkcję która dostaje string i drukuje częstość występowania wszystkich liter. Znajdź próbki tekstu w kilku językach i porównaj częstość liter. Porównaj swoje rezultaty z tabelą w https://en.wikipedia.org/wiki/Letter_frequency Całe książki w formacie tekstowym można znaleźć na <http://www.gutenberg.org/>

1.4 linie pola dipola elektrycznego

Napisz program rysujący linie pola dla dipola elektrycznego. Przykład wizualizacji pola wektorowego (dla pary rozdzielonych ładunków) jest np tutaj <https://scipython.com/blog/visualizing-a-vector-field-with-matplotlib/> lub <https://stackoverflow.com/questions/40064532/attempting-to-create-vector-field-plot-of-dipole-using-matplotlib>

1.5 symulacje błędzenia losowego

Błądzenie losowe (*random walk*) to model opisujący ewolucję jakiejś zmiennej na skutek pojedynczych losowych kroków tej zmiennej. Może być użyty do opisu zjawisk w finansach, chemii, biologii, czy fizyce polimerów. Więcej szczegółów: https://en.wikipedia.org/wiki/Random_walk lub polski odpowiednik. Wykonaj symulację błędzenia losowego na dwuwymiarowej sieci kwadratowej. Wykonaj 3 wykresy jak najbardziej podobne do tych na powyższych stronach (Wykresy ruchu ośmiu cząstek otrzymane w symulacji błędzenia losowego.), (Random walk in two dimensions with 25 thousands steps) oraz (Three random walks in three dimensions).

1.6 prey-predator model

Liczba futer zajęcy i rysy sprzedanych przez Hudson's Bay Company w latach 1845–1935 wykazuje wyraźnie oscylujący charakter. (patrz pierwszy wykres na stronie https://en.wikipedia.org/wiki/Lotka-Volterra_equations) Takie zależności wynikają ze zmian liczebności ofiar i drapieżników, i można to próbować symulować stosując podane na tej stronie równania różniczkowe Lotka–Volterra.

Napisz program robiący taką symulację i zrób w nim dwa wykresy, jak najbardziej zbliżone do dwu wykresów po prawej na powyższej stronie wikipedii. (Population dynamics for baboons and cheetahs problem mentioned aside.) oraz (Phase-space plot for the predator prey problem for various initial conditions of the predator population.)

Podobne symulacje i artykuł na ten temat: <https://www.digitalbiologist.com/blog/2018/9/a-population-dynamics-model-in-five-lines-of-python>

1.7 wielomiany Czebyszewa

Znajdź na wikipedia.org co to są wielomiany Czebyszewa T_k . Wykorzystaj rekurencyjną definicję tych wielomianów, żeby wyliczyć współczynniki dowolnego wielomianu T_k i stwórz tablicę tych współczynników dla $k = 1, \dots, 20$. Wypisz te wielomiany w czytelnej postaci i porównaj z podanymi na wikipedii kilkoma pierwszymi, żeby sprawdzić czy program dobrze działa. Użyj otrzymanych wielomianów do zrobienia wykresów kilku $T_k(x)$ w przedziale $(-1.1, 1.1)$.

1.8 klasa wielomianów

Napisz funkcję, która wylicza wielomian o podanych współczynnikach, dla podanego argumentu. Zastanów się jak prosto reprezentować wielomian.

Następnie napisz klasę, która działa na wielomianach, tak żeby można było:

```
w1 = Wielo([1,0,3])           # stworzyć wielomian np. 1 + 3*x**2
w2 = Wielo('1+3x^2')         # to ambitniejsza wersja, niekoniecznie potrzebna
w1.val(x)                    # wyliczać wartość dla podanego x
w1.stopien                   # dostać stopień wielomianu
w4 = w1+w2                   # dodawać wielomiany
w5 = w1*w2                   # mnożyć wielomiany
w6 = 4*w4+w5                 # mnożyć przez liczbę
str(w6)                      # zwraca string z wielomianem w czytelnej formie, np. '1 + 3x^2'
```

1.9 funkcja logistyczna i chaos

Przedstaw na wykresie wartości 1000 kolejnych x_n dla iteracyjnie wyliczanej funkcji logistycznej, tzn. $a = 3.9$, $x_{n+1} = a * x_n * (1 - x_n)$. Jak to zachowanie zależy od a ? Narysuj wykres

wartości x_n w funkcji a . Dalsze informacje: https://en.wikipedia.org/wiki/Logistic_map
https://upload.wikimedia.org/wikipedia/commons/0/0a/Subsection_Bifurcation_Diagram_Logistic_Map.png

1.10 kryształ NaCl

Napisz program wyliczający energię wiązania w kryształ jonowym NaCl (https://pl.wikipedia.org/wiki/Sta%C5%82a_Madelunga). Program powinien generować trójwymiarową sieć ładunków (ich współrzędne), następnie posumować energię oddziaływania elektrostatycznego środkowego jonu z pozostałymi. Wynik może zależeć od rozmiaru i kształtu generowanego kryształu. Proszę to sprawdzić.

1.11 mapa świata

Plik <http://www.zinm.if.uj.edu.pl/documents/5807277/144533431/altitude.txt> zawiera mapę wysokości i głębokości na kuli ziemskiej na prostokątnej siatce 1024x512 (są to przerzedzone dane ze strony <http://www.ngdc.noaa.gov/mgg/global/global.html>) Na podstawie tych danych: 1) policz jaki procent stanowią lądy. 2) Jaki procent lądów jest na półkuli północnej. 3) Ile wody jest w morzach i oceanach. Potrzebne wzory można znaleźć na https://www.ngdc.noaa.gov/mgg/global/etopo1_ocean_volumes.html, można korzystać również z uproszczonych wzorów bez brania pod uwagę odchylenia kształtu Ziemi od kuli.

Można również narysować z tych danych mapę świata, choć dla satysfakcjonującego rezultatu trzeba opanować używanie palet kolorów. Gotowe palety uwydatniające rzeźbę terenu na lądach lub morzach: <http://soliton.vm.bytemark.co.uk/pub/cpt-city/ngdc/tn/ETOP01.png.index.html>

1.12 seriale

Zrób wykres przedstawiający średnią ocenę widzów dla poszczególnych odcinków ulubionego serialu. Można zobaczyć tego typu wykresy <https://i.redd.it/h3yxb4tmvgp41.png> opublikowane na forum <https://www.reddit.com/r/dataisbeautiful/>

Średnie oceny różnych filmów są podane w Internet Movie Database imdb.com Wyniki można ściągnąć w formacie tekstowym ze strony <https://datasets.imdbws.com/> Są to duże pliki tekstowe, krótki opis formatu jest na <http://www.imdb.com/interfaces/>

Należy ręcznie ściągnąć te pliki na swój komputer i rozpakować (być może trzeba pozmienić nazwy po rozpakowaniu). Dalej, przy pomocy skryptu w Pythonie należy:

- w pliku `title.basics.tsv` znaleźć kod danego serialu,
- w pliku `title.episode.tsv` znaleźć kody poszczególnych odcinków,
- w pliku `title.ratings.tsv` znaleźć oceny poszczególnych odcinków,
- te dane warto zebrać tymczasowo w małą tablicę do dalszych testów,
- zrobić z tych danych wykres.

Na koniec trzeba zrobić taką funkcję, która dostaje jako parametr tytuł serialu i robi automatycznie wykres, n.p. `plot_ratings('Breaking Bad')`

