

Ćwiczenia 1

1. Załóż konto na Githubie (o ile nie masz), załóż tam prywatne repozytorium o nazwie PO123456 gdzie 123456 to Twój numer indeksu.
2. Dodaj do repozytorium jako współpracownika użytkownika o nicku pjastr
3. Skonfiguruj notatnik jupitera w projekcie w Pycharmie.
4. Przypomnij informacje o strukturach danych
 - <https://docs.python.org/3/library/stdtypes.html#sequence-types-list-tuple-range>
 - <https://docs.python.org/3/tutorial/datastructures.html>
5. Stwórz listę 10-elementową zawierającą liczby całkowite (typ `int`). Następnie poćwicz operacje przedstawione w części teoretycznej.
6. Stwórz listę 10-elementową zawierającą liczby wymierne (typ `float`). Następnie poćwicz operacje przedstawione w części teoretycznej.
7. Stwórz listę 10-elementową zawierającą napisy (typ `str`). Następnie poćwicz operacje przedstawione w części teoretycznej.
8. Stwórz listę 10-elementową zawierającą elementy różnych typów. Następnie poćwicz operacje przedstawione w części teoretycznej.
9. Stwórz listy 20-elementowe zawierającą następujące elementy:
 - a) kolejne liczby parzyste od 2 w górę
 - b) naprzemiennie 3 4 3 4 ...
 - c) naprzemiennie -1 1 -1 1 ...
 - d) kolejne potęgi trójki 3,9,27,...
10. Napisz program obliczający sumę dwóch liczb całkowitych. W programie stwórz własną metodę z dwoma parametrami i zwracającą sumę. Napisz trzy przypadki testowe.
11. Napisz program do obliczenia silni. Nie korzystaj z gotowych funkcji, napisz własną funkcję w której wnętrzu wykorzystasz pętlę. Napisz trzy przypadki testowe.
12. Napisz program do obliczenia silni. Nie korzystaj z gotowych funkcji, napisz własną funkcję rekurencyjną.
13. Zdefiniuj funkcję "cg", która dla podanych trzech parametrów: n=numer elementu ciągu, a1=wartość pierwszego elementu ciągu (domyślnie 1), q=wartość iloczynu ciągu geometrycznego (domyślnie 2) zwróci n-ty element ciągu geometrycznego. https://pl.wikipedia.org/wiki/Ci%C4%85g_geometryczny. Stwórz dwa przypadki testowe.
14. Napisz program i funkcję, która sprawdza czy dane słowo jest palindromem (brzmi tak samo czytane od lewej do prawej i od prawej do lewej). Stwórz dwa przypadki testowe.
15. Napisz program z funkcją, która sprawdza czy na liście elementy są posortowane od największych do najmniejszych (wtedy wyrzuca True, w przeciwnym wypadku False). Stwórz dwa przypadki testowe.
16. Napisz program, która zlicza sumę kwadratów elementów ciągu podanych jako parametr funkcji (użyj * do oznaczenia dowolnej ilości parametrów). Stwórz dwa przypadki testowe.
17. Napisz program z funkcją z parametrem kwargs (użyj symbolu **). Parametrem ma być słownik, którego kluczem ma być imię, a wartością wiek osoby. Funkcja ma zwracać średnią wieku osób. Stwórz dwa przypadki testowe.

18. Napisz program z funkcją z parametrem z *. Funkcja ma zwracać ciąg wejściowy podniesiony do trzeciej potęgi. Nie korzystaj z pętli, użyj tzw. List Comprehensions we wnętrzu funkcji. Stwórz dwa przypadki testowe.
19. Napisz funkcję, która zlicza ile razy została wywołana. Wskazówka: link. Stwórz 3 przypadki testowe.
20. Napisz funkcję `minmax`, która zwraca jako swój wynik krotkę dwóch liczb, z których pierwsza to minimum a druga to maksimum z listy podanej jako jedyny argument tej funkcji. Przykładowo, dla listy `a = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]` funkcja `minmax` powinna zwrócić krotkę `(28, 31)`. Przetestuj tę funkcję w funkcji `main`.
21. Napisz program, który dodaje wszystkie liczby od 2 do 1000 do początkowo pustej listy, po czym usuwa wielokrotności 2 (ale nie 2), wielokrotności 3 (ale nie 3), i tak dalej, aż do wielokrotności 100. Następnie program wypisuje liczby pozostałe na liście.

Wygeneruj sobie listę 200 elementów z zakresu od 1 do 100. Jako `seed` ustaw swój numer indeksu.

Przykładowy kod:

```
import random

random.seed(123456)
randomlist = [random.randrange(1, 101) for i in range(200)]
```

Na bazie tej listy wykonaj ćwiczenia:

21. Napisz program, który obliczy i wypisze na ekran sumę wszystkich elementów listy.
22. Napisz program, który znajdzie i wypisze na ekran najmniejszy element listy (wykonaj 2 wersje, bez użycia i używając wbudowaną funkcję języka Python).
23. Napisz program, który znajdzie i wypisze na ekran największy element listy (wykonaj 2 wersje, bez użycia i używając wbudowaną funkcję języka Python).
24. Napisz program, który znajdzie i wypisze na ekran medianę elementów listy.
25. Napisz program, który posortuje elementy listy od najmniejszego i wypisze na ekran pierwsze 20 z nich (wykonaj 2 wersje, bez użycia i używając wbudowaną funkcję języka Python).
26. Napisz program, który obliczy i wypisze na ekran iloczyn wszystkich elementów listy.
27. Napisz program, który sprawdzi i wypisze na ekran ilość liczb 3-cyfrowych na liście
28. Napisz program, który sprawdzi i wypisze na ekran liczbę oraz jej ilość powtórzeń, która najczęściej występuje na liście.
29. Napisz program, który sprawdzi i wypisze na ekran wszystkie liczby, które się nie powtarzają na liście.
30. Napisz program, który sprawdzi i wypisze na ekran, które liczby powtarzają się na liście dokładnie 3 razy.
31. Napisz program, który sprawdzi i wypisze na ekran ile liczb jest większych niż 27.
32. Napisz program, który sprawdzi i wypisze na ekran trzy najmniejsze i trzy największe liczby z listy.
33. Napisz program, który usunie z listy duplikaty, następnie wypisze na ekran ilość pozostałych elementów.
34. Wypisz wszystkie elementy listy, umieszczając przy każdym informację, ile razy występuje na liście.
35. Oblicz ile jest elementów listy, których wartość mieści się w przedziale $\langle a, b \rangle$. Wartości `a` i `b` podaje użytkownik.
36. Oblicz ile jest elementów listy, których wartość jest parzysta.
37. Oblicz ile jest elementów listy, których wartość jest nieparzysta.
38. Napisz program, który przeniesie wszystkie elementy parzyste listy posegregowane od najmniejszego na lewą stronę listy, natomiast nieparzyste posegregowane malejąco na prawą. Np. `[2,1,4,6,3,6,5]` -> `[2,4,6,6,5,3,1]`

Dalsze zadania:

39. Napisz program, w którym znajduje się słownik z minimum 5 elementami. Posortuj słownik wg kluczy, a następnie wg wartości (załóż że wykorzystany typ kluczy i wartości umożliwia sortowanie).

40. Napisz skrypt, który dwa słowniki zostaną połączone w nowy słownik. W przypadku duplikatów kluczy, usuń wartość z drugiego słownika.
41. Napisz program, który zsumuje wszystkie wartości ze słownika (załóż, że wartości w słowniku można sumować).

Zrób dwa warianty, gdy wartości są liczbowe lub są napisami.

42. Napisz program, który zamieni całkowitą liczbę dziesiętną na odpowiadającą jej liczbę rzymską. Wykorzystaj w tym celu odpowiednik słownik.
43. Napisz program, który usuwa jeden element z krotki (załóż, że krotka ma co najmniej 3 elementy).
44. Napisz program, który zamienia krotką składającą się z dodatnich liczb całkowitych z zakresu 1-9 na liczbę całkowitą.

Przykład: krotka (4,8,9) ma zostać zamieniona na liczbę 489.

45. Za pomocą konstrukcji m.in. tzw. Python comprehension stwórz krotkę zawierającą liczby parzyste między 2 a 50 (włącznie).
46. Napisz program, który wpisuje wszystkie elementy zbioru.
47. Napisz program, który łączy dwa zbiory w jeden.
48. Napisz program, który oblicza część wspólną dwóch zbiorów.
49. Napisz program (bez gotowej funkcji), które oblicza liczbę elementów w zbiorze.