

Listy i krotki. Obiekty iterowalne. Pętle.

Zadanie 2.1. (MG)

Przy użyciu pętli `for` lub `while` oraz funkcji `print()` wypisz na konsoli następujące wartości:

- dla $i = 0, 1, \dots, n - 1$ wypisz i (dla pewnego $n \in \mathbb{N}$),
- dla $i = n, n - 1, \dots, 0$ wypisz i (dla pewnego $n \in \mathbb{N}$),
- dla $j = 1, 3, \dots, 2k - 1$ wypisz j (dla pewnego $k \in \mathbb{N}$),
- dla $i = 1, 2, 4, 7, \dots, n$ wypisz i (dla pewnego $n \in \mathbb{N}$),
- dla $j = 1, 2, 4, 8, 16, \dots, n$ wypisz j (dla pewnego $n \in \mathbb{N}$),
- dla $j = 1, 2, 4, 8, 16, \dots, 2^k$ wypisz j (dla pewnego $k \in \mathbb{N}$),
- dla $x = a, a + \delta, a + 2\delta, \dots, b$ wypisz x (dla pewnych $a, b, \delta \in \mathbb{R}, a < b, \delta > 0$),
- dla $i = 2, 4, 6, 8, 12, \dots$ wypisz i (podzielna przez 2 ale niepodzielna przez 10, mniejsza niż 100),
- dla $i = 5, 10, 15, 21, 28, \dots$ wypisz i (co druga podzielna przez 5 lub przez 7, mniejsza niż 100).

Stałe N , a, b oraz δ wczytaj przy użyciu funkcji `input()`.

Zadanie 2.2.

Przy użyciu pętli `for` oraz funkcji `print()` wypisz na konsoli:

- wszystkie litery słowa `Latajacy cyrk Monty Pythona`,
- każdą literę słowa `Latajacy cyrk Monty Pythona` oraz jej pozycję,
- kombinację `imie: nazwisko:` dla napisów zawartych w listach

```
Imiona = ["Anna", "Barbara", "Robert", "Adam"]
Nazwiska = ["Kowalski", "Smith", "Malinowski", "Grzegorzówka"]
```

- pierwszy element każdej krotki z następującej listy `[(1, 2, 3), (4, 5, 6), (7, 8, 9)]`,

Zadanie 2.3.

Pokaż w jaki sposób przy użyciu różnych sposobów można stworzyć listę, której elementami będą kolejne liczby całkowite $0, 1, 2, \dots, n - 1$ dla zadanego $n \geq 20$. Wykorzystaj

- metodę `.append()`;
- listy składane;
- funkcję `list()`.

Następnie dla utworzonej listy:

- wybierz pierwszy i ostatni element;
- wybierz wszystkie elementy z wyjątkiem pierwszego;
- wybierz wszystkie elementy z wyjątkiem ostatniego;
- wybierz co trzeci element poczynając od elementu 4. aż do elementu 10.;
- dodaj element o wartości -10 na pozycję 5;
- usuń ostatni element;
- usuń element o wartości 7;
- wszystkie elementy parzyste zamień na kwadraty ich wielkości;
- dodaj do listy elementy `[100, 101, 102]`;
- powtórz każdy element z listy 3 krotnie.

Zadanie 2.4.

Utwórz listę składającą się

- z wszystkich liczb parzystych ze zbioru $\{0, 1, 2, \dots, 10\}$;
- z kwadratów wszystkich liczb ze zbioru $\{0, 1, 2, \dots, 10\}$;
- z kwadratów wszystkich liczb parzystych ze zbioru $\{0, 1, 2, \dots, 10\}$;

- z sum elementów dwóch list o wartościach $[0, 1, 2, 3, 4, 5]$ oraz $[100, 200, 300, 400, 500]$;
- z sum parzystych elementów dwóch list o wartościach $[0, 1, 2, 3, 4, 5]$ oraz $[100, 200, 300, 400, 500]$;
- z sum odpowiadających sobie elementów dwóch list o wartościach $[1, 2, 3, 4, 5]$ oraz $[100, 200, 300, 400, 500]$;
- napisów będących sklejonymi literami napisów JAKO oraz jajko.

W rozwiązaniu wykorzystaj zarówno listy składane, a także znane Ci inne sposoby tworzenia list.

Uwaga Wróć do tego zadania po wykładzie nt. funkcji. Przestudiuj dokumentację dla funkcji `map()` i `filter()` i spróbuj wykonać polecenia z punktów (a) – (d) również przy ich użyciu.

Zadanie 2.5.

Utwórz macierz $M \in \mathbb{R}^{n \times m}$ o elementach całkowitych. Macierz będziemy reprezentowali przy użyciu zagnieżdżonych list, np.

```
[ [1, 2, 3],
  [4, 5, 6],
  [7, 8, 9] ]
```

- Utwórz listę składającą się z elementów pierwszego wiersza macierzy M .
- Utwórz listę składającą się z elementów drugiej kolumny macierzy M .
- Utwórz listę składającą się z elementów diagonal macierzy M .
- Dla macierzy $N \in \mathbb{R}^{m \times n}$ wykonaj mnożenie macierzy MN . Wynik powinien być w postaci zagnieżdżonej listy.

Zadanie 2.6.

Dla zadanej liczby całkowitej $n < 10$ wyznacz i wpisz na konsoli tabliczkę mnożenia $i * j$ dla $i, j \in \{1, 2, \dots, n\}$. Na przykład dla $n = 9$ powinieneś otrzymać:

	1	2	3	4	5	6	7	8	9
1	1	2	3	4	5	6	7	8	9
2	2	4	6	8	10	12	14	16	18
3	3	6	9	12	15	18	21	24	27
4	4	8	12	16	20	24	28	32	36
5	5	10	15	20	25	30	35	40	45
6	6	12	18	24	30	36	42	48	54
7	7	14	21	28	35	42	49	56	63
8	8	16	24	32	40	48	56	64	72
9	9	18	27	36	45	54	63	72	81

Zadanie 2.7. (MG)

Napisz program, który wczyta kolejne cyfry liczby całkowitej nieujemnej zapisanej w notacji ósemkowej oraz obliczy i wypisze jej wartość w notacji dziesiętnej.

- Program powinien wczytywać cyfry pojedynczo, w kolejności od najmniej znaczącej do najbardziej znaczącej cyfry. Znaczy to, że wpisanie trzech cyfr w kolejności 3, 2, 1 oznacza wprowadzenie liczby 123 (ósemkowo), co powinno dać w wyniku (na konsoli) 83 w notacji dziesiętnej.
- Program powinien zakończyć wczytywanie kolejnych cyfr po podaniu 21 cyfr lub gdy wczytana wartość nie jest cyfrą w systemie ósemkowym.

Zadanie 2.8.

Zaimplementuj samodzielnie proste algorytmy sortowania:

- bąbelkowe,
- przez wstawianie,
- przez wybór,

służące do sortowania danej listy w miejscu.

Zadanie 2.9. (MG)

Dana jest następująca funkcja (sposób jej działania narazie nas nie interesuje):

```
import scipy.stats as stats
import math

def F(a, b):
    """
    Nie interesuje nas, co funkcja robi:
    traktujemy ja jako "czarna skrzynkę".

    Ważne jest jedynie to, że F(a, b) zwraca wartość z przedziału [0,1]
    dla a, b > 0
    """
    return stats.beta.cdf(0.5, a=a, b=b)
```

Przetestuj metodę poszukiwanie po siatce (ang. *{grid search}*) wartości funkcji. Polega ono na obliczaniu wartości $F(a_i, b_j)$ dla wszystkich kombinacji wyrazów dwóch ciągów arytmetycznych $(a_i)_{i=1,2,\dots,n}$ oraz $(b_j)_{j=1,2,\dots,m}$ (tworzą one w istocie siatkę równoodległych punktów).

- Wczytaj wartości n , m , a_1 , a_n , b_1 , b_m .
- Wyznacz minimum i maksimum funkcji w punktach siatki.
- Wyznacz wartości funkcji F w każdym punkcie z siatki i wypisz wyniki w postaci estetycznie sformatowanej tabelki, np. dla wartości:

```
a_1 = 0.5
a_n = 5
n = 10
b_1 = 1
b_m = 7
m = 8
```

powinniśmy uzyskać wynik:

a / b	1.00	1.86	2.71	3.57	4.43	5.29	6.14	7.00
0.5	0.71	0.87	0.94	0.97	0.98	0.99	1.00	1.00
1.0	0.50	0.72	0.85	0.92	0.95	0.97	0.99	0.99
1.5	0.35	0.59	0.75	0.85	0.91	0.95	0.97	0.98
2.0	0.25	0.47	0.64	0.77	0.85	0.91	0.94	0.96
2.5	0.18	0.37	0.54	0.68	0.78	0.86	0.91	0.94
3.0	0.12	0.28	0.45	0.59	0.71	0.80	0.86	0.91
3.5	0.09	0.22	0.37	0.51	0.64	0.74	0.81	0.87
4.0	0.06	0.17	0.30	0.43	0.56	0.67	0.76	0.83
4.5	0.04	0.13	0.24	0.37	0.49	0.60	0.70	0.78
5.0	0.03	0.10	0.19	0.30	0.42	0.54	0.64	0.73