

## WSTĘP DO PROGRAMOWANIA

### Laboratorium 5

#### Listy i krotki

Zadania przygotowały: dr Anna Nenca, Laura Grzonka

#### Listy

**Listy** to to jedna ze struktur danych. Ich najistotniejsze właściwości to:

- są **indeksowalne**;
- są **modyfikowalne**;
- **mogą zawierać duplikaty**.

Zad. 1. Sprawdź, co robią następujące programy:

a)

```
napis="Ala ma kota"
print(napis[0])
print(napis[2])
print(napis[1:3])
print(napis[1::3])
print(napis[::])
print(len(napis))
print(napis.lower())
print(napis.count(" "))
```

b)

```
lista1=[1,2,3,4,5,6]
lista2=[1,3,4,5,9]
print(lista1,lista2)
lista3=[lista1,lista2]
lista4=lista1+lista2
print(lista3)
print(lista4)
```

c)

```
lista1=[1,2,3,4,5,6]
for i in lista1:
    print(i)
for i in range(len(lista1)):
    print(lista1[i])
```

d)

```
import random
print(random.randint(3, 9))
```

e)

```
import random
lst = []
for i in range(10):
    lst.append(random.randint(1,10))
```

```
# Prints random items
print(lst)
```

Zad. 2. Napisz program, który policzy liczbę samogłosek w zadanym słowie. Zwróć uwagę na małe, duże litery.

Zad. 3. Utwórz listę `oceny = [5, 3, 4, 2, 5, 1]` i posortuj ją rosnąco, a następnie wyświetl wynik.

Zad. 4. Mając listę `zakupy = ["mleko", "chleb", "masło", "ser"]`, sprawdź, czy "jajka" znajdują się na liście. Wyświetl `True` lub `False` w zależności od wyniku.

Zad. 5. Napisz program, który znajdzie najmniejszą i największą wartość należącą do listy liczb.

Zad. 6. Napisz program, który znajdzie element o najmniejszej wartości należący do listy i wypisze jego pozycję na liście.

Zad. 7. Napisz program, który znajdzie drugi największy element na liście i wypisze jego pozycję na liście, np. dla listy `[4, 7, 2, 9, 6, 8]` będzie to wartość: 8, indeks: 5.

Zad. 8. Napisz program, który wypisze wszystkie napisy należące do listy dłuższe niż 3 znaki.

Zad. 9. Napisz program, który wypisze `TAK`, jeśli dwie listy mają wspólny element, lub `NIE`, jeżeli nie mają wspólnych elementów.

Zad. 10. Napisz program, który wypisze tylko te liczby z listy, które są nieparzyste.

Zad. 11. Napisz program, który dla liczb mniejszych od tysiąca wyświetli ich wartość słownie, np. 372=trzysta siedemdziesiąt dwa.

Zad. 12. Napisz program, który dla dowolnego słowa i wzorca sprawdzi czy dany wzorec występuje w słowie, np. słowo: rabarbar, wzorec: ab odpowiedź `TAK`. Rozwiąż to zadanie bez korzystania wbudowanych metod.

Zad. 13. Napisz program, który sprawdzi czy podany napis jest palindromem.

Zad. 14. Napisz program, który z podanej listy liczb wypisze te elementy, które mają parzyste indeksy. Pamiętaj, że zaczynamy numerowanie indeksów od zera.

Zad. 15. Napisz program, który z podanej listy napisów utworzy inną listę zawierającą pary (napis, długość napisu). Zwróć uwagę na nawiasowanie.

Zad. 16. Napisz program, który wypisze odpowiedź na pytanie, czy z góry zadany element pojawił się na liście, np. dla 4 i listy `[1, 5, 3, 6, 4]` odpowiedzią jest `TAK`.

Zad. 17. Napisz program, który dla listy składającej się z  $n$  list z których każda ma  $m$  elementów, wyświetli zawartość każdego wiersza w odwrotnej kolejności.

Zad. 18. Napisz program, który zamieni miejscami pierwszy oraz ostatni element zadanej listy miejscami. Wyświetl listę przed i po zamianie. Nie używaj przy tym operacji zmieniających długość listy; nie używaj dodatkowych list.

Zad. 19. Napisz program, który dla danego  $n$  oraz dla listy zagnieżdżonej o wymiarach  $n \times n$  zawierającej liczby rzeczywiste, wyświetli sumę wartości liczb, które znajdują się na przekątnych tej macierzy (macierzy  $n \times n$ ).

Zad. 20. Napisz program, który realizuje mnożenie macierzy przez skalar:

np. **wejście:** `[[1, 2, 4, 6], [2, 3, 4, 5], [12, 3, 4, 5]], 3`

daje **wyjście**: `[[3, 6, 12, 18], [6, 9, 12, 15], [36, 9, 12, 15]]`.

**Zad. 21.** Napisz program, który uzupełni listę dwudziestoma kolejnymi liczbami naturalnymi, a następnie używając ciągu operacji na listach utworzy następującą listę zagnieżdżoną: `[[1, 2, ..., 20], [20, 19, ..., 1]]`.

**Zad. 22.** Napisz program, który w miejsce ostatniego elementu danej listy wstawi wszystkie elementy z innej listy:

np. dla danych `[1, 3, 5, 7, 9, 10]`, `[2, 4, 6, 8]` wynik to `[1, 3, 5, 7, 9, 2, 4, 6, 8]`.

**Zad. 23.** Napisz program, który z listy list wybierze listę o najwyższej oraz o najniższej sumie.

**Zad. 24.** Napisz program, który dla danej macierzy wypisze macierz transponowaną.

**Zad. 25.** Napisz program, który wypisze drugą największą wartość z listy.

**Zad. 26.** Napisz program, który wygeneruje losową listę liczb, ustawioną w porządku od najmniejszej wartości do największej. Nie używaj przy tym sortowania.

**Zad. 27.** Napisz program, który połączy dwie posortowane listy w jedną, także posortowaną, ale nie używaj sortowania.

**Zad. 28.** Wyświetl niepowtarzające się wartości z listy list:

np. dla `[[1, 2, 3, 5], [2, 3, 5, 4], [0, 5, 4, 1], [3, 7, 2, 1], [1, 2, 1, 1, 2]]`

mamy `[0, 1, 2, 3, 4, 5, 7]`.

**Zad. 29.** Stwórz program, który wykona przesunięcie cykliczne listy:

np. dla `['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h']` przesunięcie o 3 daje `['d', 'e', 'f', 'g', 'h', 'a', 'b', 'c']`.

## Krotki

**Krotki** to to jedna ze struktur danych. Ich najistotniejsze właściwości to:

- są **indeksowalne**;
- **nie** są **modyfikowalne**;
- **mogą zawierać duplikaty**.

**Zad. 1.** Mając krotkę `kolory = ("czerwony", "zielony", "niebieski", "żółty")`, wyświetl jej pierwszy oraz ostatni element.

**Zad. 2.** Mając krotkę `dni_tygodnia = ("poniedziałek", "wtorek", "środa", "czwartek", "piątek", "sobota", "niedziela")`, sprawdź, czy znajduje się w niej "poniedziałek". Wyświetl `True` lub `False` w zależności od wyniku.

**Zad. 3.** Mając dwie krotki `krotka1 = (1, 2, 3)` i `krotka2 = (4, 5, 6)`, połącz je w jedną krotkę i wyświetl wynik.

**Zad. 4.** Do krotki `kraje = ("Polska", "Niemcy", "Francja")` dodaj element "Hiszpania".

**Zad. 4.** Kiedy używamy krotek, a kiedy list?