

# Wstęp do Informatyki 2023/2024

## Lista #10

Instytut Informatyki, Uniwersytet Wrocławski

19, 22 grudnia 2023 i 10 stycznia 2024

Zdefiniowany został typ `struct elem` w C oraz klasa `ListItem` w Pythonie.

```
struct elem {  
    int val;  
    struct elem* next;  
};
```

```
class ListItem:  
    def __init__(self, value):  
        self.val = value  
        self.next = None
```

W rozwiązaniach zadań w języku C można korzystać z funkcji `struct elem* utworz(int value)` tworzącej listę jednoelementową z kluczem `wart` (p. slajdy do wykładu). W Pythonie listę jednoelementową tworzymy przy pomocy klasy `ListItem` z polami `val` i `next` wywołując konstruktor `ListItem(wart)`.

Napisz funkcje realizujące następujące operacje:

1. [0.5] Dołączenie nowego elementu na koniec listy.
2. [0.5] Usunięcie ostatniego elementu z listy.
3. [1] Dołączenie jednej listy na koniec drugiej.
4. [1] Usunięcie z listy **wszystkich** elementów o podanej wartości pola `val`. Złożoność czasowa Twojego rozwiązania powinna być  $O(n)$ , gdzie  $n$  to liczba elementów na liście.
5. [1] Zaproponuj sposób reprezentacji listy (jednokierunkowej), który umożliwi realizację operacji z zadań 1 i 3 oraz operacji wstawiania/usuwania elementu z początku listy w czasie  $O(1)$ . Podaj jak zaimplementować te operacje w nowej reprezentacji.  
*Uwaga.* Dodatkowy koszt pamięciowy związany z nową reprezentacją powinien być  $O(1)$ .
6. [1] Napisz funkcję wypisującą na standardowym wyjściu elementy listy w odwrotnej kolejności do ich występowania w liście wejściowej. Nie należy przy tym zmieniać kolejności elementów w liście ani tworzyć nowej listy.  
*Wskazówka.* Wykorzystaj następującą obserwację: aby wypisać od końca elementy niepustej listy  $L$ , wystarczy najpierw wypisać od końca elementy  $L$  bez pierwszego elementu, a potem wypisać pierwszy element  $L$ .
7. [1] Napisz funkcję umożliwiającą odwrócenie kolejności elementów na liście jednokierunkowej. W Twojej implementacji nie powinny być tworzone nowe elementy listy a jedynie zmieniane wskaźniki.

8. [1] Napisz funkcję, która rozdzieli daną listę na dwie podlisty: jedną zawierającą elementy z kluczami dodatnimi a drugą – elementy z kluczami ujemnymi. W Twojej implementacji nie powinny być tworzone nowe elementy listy a jedynie zmieniane wskaźniki.
9. [2] Zaproponuj typ danych dla elementów listy dwukierunkowej, tj. takiej, w której każdy element zawiera wskaźnik na następny i wskaźnik na poprzedni element w liście (elementy pierwszy i ostatni mają odpowiednie wskaźniki ustawione na NULL/None). Napisz funkcje realizujące operacje kolejkowe na takiej liście (dodanie elementu na koniec kolejki, usunięcie elementu z początku kolejki) a także usunięcie elementu z końca listy.
10. [1] Napisz funkcję, która scala dwie **uporządkowane** listy dając również w wyniku listę uporządkowaną. W Twojej implementacji nie powinny być tworzone nowe elementy listy a jedynie zmieniane wskaźniki.

Zadania dodatkowe, nieobowiązkowe (nie wliczają się do puli punktów do zdobycia na ćwiczeniach, punktacja została podana tylko jako informacja o trudności zadań wg wykładowcy):

1. [0.5] Napisz funkcję wypisującą na standardowym wyjściu wszystkie dodatnie elementy z listy.
2. [0.5] Przedstaw sposób implementacji stosu przy pomocy list, czyli podaj funkcje realizujące operacje: wstaw element na szczyt stosu, usuń element ze szczytu stosu, zainicjuj (pusty) stos.
3. [0.5] Przedstaw sposób implementacji kolejki przy pomocy list, czyli podaj funkcje realizujące operacje: wstaw element na koniec kolejki, usuń element z początku kolejki, zainicjuj (pustą) kolejkę.
4. [1] Napisz funkcje pozwalające usuwać i dodawać elementy do **uporządkowanej** listy dwukierunkowej.