

Задачи:

1. Реализирайте следните методи за свързан списък:

- count()
- getNth()
- deleteList()
- pop()
- insertNth()
- reverse()
- recursiveReverse()
- finding the middle node (***hint** use two pointers)
- append(LList<T> a, LList<T> b) – присъединява 'b' в края на 'a' и set-ва 'b' на NULL
- detectCycle() – проверява дали в свързаният списък има цикъл

2. Flattening a Linked List

Даден ви е свързан списък, в който всеки елемент представлява Node с 2 указателя.

- Първият указател представлява указател към следващия елемент в основният Linked List ('right' pointer)
- Вторият указател представлява началото на друг Linked List ('down' pointer)

*Всички свързани списъци са сортирани

Пример :

5 -> 10 -> 19 -> 28

| | | |

v v v v

7 20 22 35

| | |

v v v

8 50 40

| |

v v

30 45

Изправеният свързан списък също трябва да е сортиран. Резултат от примера: 5->7->8->10->19->20->22->28->30->35->40->45->50

3. Реализирайте функция, която завърта свързан списък по посока на часовниковата стрелка с k позиции. k е естествено число, по-малко от дължината на свързания списък.

Пример:

10->20->30->40->50->60 k = 4

Резултат:

50->60->10->20->30->40

4. Реализирайте функция, която проверява дали даден свързан списък е палиндром. Функцията да връща true, ако списъкът е палиндром и false – в противен случай.
5. Да се дефинира шаблон на булева функция **dupsb**, която приема два параметъра: указател към линеен едносвързан списък и неотрицателно число **k**. Функцията да връща истина, ако в списъка има поне два различни подсписъка с дължина **k**, които са с еднакви елементи. Примери за такива списъци са 1, 2, 3, 4, 1, 2, 5 и 1, 2, 3, 3, 3, 4 за **k = 2**.
6. Да се дефинира функция **removedups**, която приема като единствен параметър указател към линеен едносвързан списък от цели числа, наредени в ненамаляващ ред. Функцията да изтрива всички повторения на елементи в списъка, така че всеки от оригиналните му елементи да се среща само веднъж, така че елементите на списъка да образуват строго растяща редица. Задачата да се реши без използване на помощна структура от данни (**с константна сложност по памет**).
7. Да се дефинира шаблон на функция **removesub**, която приема два параметъра **first** и **second** — указатели към линейни едносвързани списъци. Функцията да изтрива от списъка с начало **first** последователно всички срещания на подсписъци, които съвпадат със списъка с начало **second**. Например, при списък **first** с елементи 1, 2, 3, 3, 3, 4, 5, 3, 3, 6 и **second** с елементи 3, 3, **first** да се преобразува до списъка 1, 2, 3, 4, 5, 6.