Задача 1. Нека е даден списък L с N елемента. Да се дефинира подходящо параметризирана функция *shuffle*, която получава адреса на първия елемент на списъка. Функцията да пренарежда възлите на списъка така, че елементите от втората половина на списъка да се преместят в началото на списъка, но в обратен ред (при списъци с нечетен брой елементи считаме средния елемент за принадлежащ към първата половина на списъка).

Пример:

L1 → L2 → L3 → L4 → L5 се преобразува до L5 → L4 → L1 → L2 → L3

При решението на задачата да не се изтриват или заделят нови възли, а да се използват съществуващите. Могат да се използват други изучени структури от данни.

Задача 2. Нека е даден следният шаблон на структура:

```
template <class T>

struct box{T data; box<T> *next;};
```

Да се реализира функция reduce, която приема два параметъра: указател към първия елемент на линеен едносвързан списък L с възли от тип boxи двуместна функция F от тип (const T&, const T&) → T. Резултатът от изпълнението на reduce да е стойността при приложението на ляво-асоциативния оператор F последователно над елементите на L, или F(...F(F(l1,l2), l3) ...,lk), където l1,...,lk са елементите на списъка L.

При K = 0 да се генерира подходяща грешка (изключение), а при K = 1 стойността на функцията да е l1.

Задача 3
При посочената дефиниция на линеен едносвързан списък от предишната задача, да се напишат:

- функция removeAll (box*& l,int x), която изтрива всички срещания на елемента x от списъка l.
- функция bool duplicates (box *l), която проверява дали в списъка l има дублиращи се елементи.
- функция void removeduplicates (box *&l), която изтрива всички дублиращи се елементи от списъка l.
- фунцкия bool palindrom (box *l), която проверява дали редицата от елементи на даден списък обрзува палиндром (т.е. дали се чете еднакво както отляво надясно така и отдяно наляво).

Задача 4. Нека е дадена следната структура:

```
struct Node {int data; Node<int> *next;};
```

Реализирайте функция `inc(Node* list)`, която приема като параметър указател към първия елемент на свързан списък с четен брой елементи, чиито елементи са цели числа в интервала [0..9]. Функцията да изважда единица от десетичното число, което е представено в първата половина на списъка, и да добавя единица към десетичното число, което е представено във втората половина. Функцията да връща указател към началото на списъка.

Пример: Списъкът

```
1->9->0->0->9->9->9->9
```

се трансформира до списъка:

```
1->8->9->9->1->0->0->0->0
```

Обяснение:

Числото в първата половина е 1900 - 1 = 1899

Числото във втората половина е 9999 + 1 = 10000