Задача 1. Нека е даден следният шаблон на структура:

```
template <class T>
struct Node {T data; Node<T> *next;};
```

Да се реализира функция *reduce*, която приема два параметъра: указател към първия елемент на линеен едносвързан списък L с възли от тип Node и 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(l_1,l_2), l_3) ...,l_k), където l_1,...,l_k са елементите на списъка L.

При K = 0 да се генерира подходяща грешка (изключение), а при K = 1 стойността на функцията да е l_1.

Пример:
Нека имаме списъка L с елементи 1024, 16, 4, 2. Нека d(x,y) = x/y. Тогава резултатът от reduce (L,d) ще бъде 8, тъй като div(div(div(1024, 16), 4), 2) = 8

Задача 2. При условията на горната задача, нека е даден списък L с елементи стекове. Възлите на L са от тип Node<std::stack<T>>> (или друга готова реализация на стек, с която разполагате). Да се дефинира подходящо параметризирана функция equalize(L), която размества елементите на стековете така, че да няма два стека в L с разлика в броя на елементите, по-голяма от 1.

Пример: Даден е списък от стекове и едно от възможните пренареждания на елементите на стековете. Разместените елементи са подчертани.

1 2 8 3 9 4 6 10 5 → 7 → 11 → 12	 3 <u>1</u> 9 <u>8</u> 4 6 10 <u>2</u> 5 → 7 → 11 → 12

Задача 3. 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.