

Задача 1.

Нека е дадена следната структура от данни:

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

Чрез нея е представен линеен едносвързан списък, който се състои от елементи от произволен тип T. Да се напише функция *[подходящ тип] compact([подходящ тип] start)*, която приема указател към началото на линеен едносвързан списък, представен чрез гореспоменатата структура. Функцията променя списъка като премахва всяка група от последователни повтарящи се елементи и оставя само по един представител.

Пример:

1 -> 2 -> 2 -> 2 -> 3 -> 4 -> 4 -> 2 -> 2 => 1 -> 2 -> 3 -> 4 -> 2

Задача 2.

Като се използва структурата на линеен едносвързан списък от предишната задача, да се напише функция *[подходящ тип] sort([подходящ тип] start, [подходящ тип] weight)*, която приема указател към началото на линеен едносвързан списък и теглова функция и сортира списъка, съобразно тегловата функция. Да не се променят стойностите на кутиите на свързания списък. Да се не използва излишно заделяне на памет.

Задача 3.

Да се реализира функция `box<int> range(int x, int y)`, която създава и връща линеен едносвързан списък с елементи от следната редица $x, x+1, x+2, \dots, y$ при $x \leq y$.

Задача 4.

Да се реализира функция `concat`, която конкатенира два линейни едносвързани списъка в нов списък.