

ЗАДАЧИ ЗА ЗАДЪЛЖИТЕЛНА  
САМОПОДГОТОВКА  
ПО  
Структури от данни и програмиране  
*Линеен едносвързан списък*

*email: kalin@fmi.uni-sofia.bg*

5 октомври 2017 г.

Следните задачи да се решат като упражнение за директно боравене с указателите и двойните кутии на линеен едносвързан списък.

1. Функция `int count(box* l, int x)`, която преброява колко пъти елементът `x` се среща в списъка с първи елемент `l`.
2. Функция `box* range (int x, int y)` която създава и връща първия елемент на списък с елементи  $x, x + 1, \dots, y$ , при положение, че  $x \leq y$ .
3. Функция `removeAll (box*& l, int x)`, която изтрива всички срещания на елемента `x` от списъка `l`.
4. Функция `void append(box*& l1, box* l2)`, която добавя към края на списъка `l1` всички елементи на списъка `l2`.
5. Функция `box* concat(box *l1, box* l2)`, който съединява два списъка в нов, трети списък. Т.е. `concat(l1, l2)` създава и връща нов списък от елементите на `l1`, следвани от елементите на `l2`.
6. Функция `map`, която прилага едноаргументна функция  $f : int \rightarrow int$  към всеки от елементите на списък.
7. Функция `reverse`, която обръща реда на елементите на списък. Например, списъкът с елементи 1, 2, 3 ще се преобразува до списъка с елементи 3, 2, 1.

8. Да се напише функция `bool duplicates (box *l)`, която проверява дали в списъка *l* има дублиращи се елементи.
9. Да се напише функция `void removeduplicates (box *&l)`, която изтрива всички дублиращи се елементи от списъка *l*.
10. Да се напише функция `bool issorted (box *l)`, която проверява дали даден списък е подреден в нарастващ или в намаляващ ред.
11. Да се напише функция `bool palindrom (box *l)`, която проверява дали редицата от елементи на даден списък обръзва палиндром (т.е. дали се чете еднакво както отляво надясно така и отдясно наляво).