Вектора
Разрешенные функции size() и push_back()
Заголовочный файл easy_list.h

1. Напишите функцию `void itc_even_index_list (const vector <int> &mass, vector <int> &mass2)`, которая принимает массив и ссылку на другой массив. Во второй массив записывает числа, которые находились на четных позициях первого (0, 2, 4...).
2. Напишите функцию `void itc_even_parts_list(const vector <int> &mass, vector <int> &mass2)`, которая принимает массив целых чисел, возвращает четные элементы вектора в виде нового массива.
3. Напишите функцию `int itc_positive_list(const vector <int> &mass)`, которая принимает вектор целых чисел, возвращает количество положительных чисел среди элементов данного вектора.
4. Напишите функцию `int itc_sl_list(const vector <int> &mass)`, которая принимает вектор целых чисел, возвращает количество элементов массива, больших предыдущего (элемента с предыдущим номером)
5. Напишите функцию `bool itc_same_parts_list(const vector <int> &mass)`, которая принимает вектор целых чисел, возвращает True если есть в массиве пара соседних элементов с одинаковыми знаками и False в противном случае.
6. Напишите функцию `void itc_rev_list(vector <int> &mass)`, которая переставляет элементы вектора в обратном порядке без использования дополнительного вектора. Возвращает перевернутый вектор.
7. Напишите функцию `void itc_rev_par_list(vector <int> &mass)`, которая переставляет соседние элементы вектора (1-й элемент поменять с 2-м, 3-й с 4-м и т.д. Если элементов нечетное число, то последний элемент остается на своем месте). Возвращает перевернутый вектор.
8. Напишите функцию `void itc_rshift_list(vector <int> &mass)`, которая циклически сдвигает элементы вектора вправо (например, если элементы нумеруются, начиная с нуля, то 0-й элемент становится 1-м, 1-й становится 2-м, ..., последний становится 0-м, то есть массив {3, 5, 7, 9} превращается в массив {9, 3, 5, 7}).
9. Напишите функцию `void itc_super_shift_list(vector <int> &mass, int n)`, которая сдвигает элементы вектора (сдвиг - циклический) на n элементов вправо, если n – положительное и влево, если отрицательное.