
1. ЗАДАЧА «ЧЕТНЫЕ ИНДЕКСЫ»

Выведите все элементы списка с четными индексами (то есть `A[0]`, `A[2]`, `A[4]`, ...).

2. ЗАДАЧА «ЧЕТНЫЕ ЭЛЕМЕНТЫ»

Выведите все четные элементы списка. При этом используйте цикл `for`, перебирающий элементы списка, а не их индексы!

3. ЗАДАЧА «БОЛЬШЕ ПРЕДЫДУЩЕГО»

Дан список чисел. Выведите все элементы списка, которые больше предыдущего элемента.

4. ЗАДАЧА «СОСЕДИ ОДНОГО ЗНАКА»

Дан список чисел. Если в нем есть два соседних элемента одного знака, выведите эти числа. Если соседних элементов одного знака нет — не выводите ничего. Если таких пар соседей несколько — выведите первую пару.

5. ЗАДАЧА «БОЛЬШЕ СВОИХ СОСЕДЕЙ»

Дан список чисел. Определите, сколько в этом списке элементов, которые больше двух своих соседей, и выведите количество таких элементов. Крайние элементы списка никогда не учитываются, поскольку у них недостаточно соседей.

6. ЗАДАЧА «НАИБОЛЬШИЙ ЭЛЕМЕНТ»

Дан список чисел. Выведите значение наибольшего элемента в списке, а затем индекс этого элемента в списке. Если наибольших элементов несколько, выведите индекс первого из них.

7. ЗАДАЧА «ШЕРЕНГА»

Петя перешёл в другую школу. На уроке физкультуры ему понадобилось определить своё место в строю. Помогите ему это сделать.

Программа получает на вход невозрастающую последовательность натуральных чисел, означающих рост каждого человека в строю. После этого вводится число X – рост Пети. Все числа во входных данных натуральные и не превышают 200.

Выведите номер, под которым Петя должен встать в строй. Если в строю есть люди с одинаковым ростом, таким же, как у Пети, то он должен встать после них.

8. ЗАДАЧА «КОЛИЧЕСТВО РАЗЛИЧНЫХ ЭЛЕМЕНТОВ»

Дан список, упорядоченный по неубыванию элементов в нем. Определите, сколько в нем различных элементов.

9. ЗАДАЧА «ПЕРЕСТАВИТЬ СОСЕДНИЕ»

Переставьте соседние элементы списка ($A[0]$ с $A[1]$, $A[2]$ с $A[3]$ и т. д.). Если элементов нечетное число, то последний элемент остается на своем месте.

10. ЗАДАЧА «ПЕРЕСТАВИТЬ MIN И MAX»

В списке все элементы различны. Поменяйте местами минимальный и максимальный элемент этого списка.

11. ЗАДАЧА «УДАЛИТЬ ЭЛЕМЕНТ»

Дан список из чисел и индекс элемента в списке k . Удалите из списка элемент с индексом k , сдвинув влево все элементы, стоящие правее элемента с индексом k .

Программа получает на вход список, затем число k . Программа сдвигает все элементы, а после этого удаляет последний элемент списка при помощи метода `pop()` без параметров.

Программа должна осуществлять сдвиг непосредственно в списке, а не делать это при выводе элементов. Также нельзя использовать дополнительный список. Также не следует использовать метод `pop(k)` с параметром.

12. ЗАДАЧА «ВСТАВИТЬ ЭЛЕМЕНТ»

Дан список целых чисел, число k и значение c . Необходимо вставить в список на позицию с индексом k элемент, равный c , сдвинув все элементы, имевшие индекс не менее k , вправо.

Поскольку при этом количество элементов в списке увеличивается, после считывания списка в его конец нужно будет добавить новый элемент, используя метод `append`.

Вставку необходимо осуществлять уже в считанном списке, не делая этого при выводе и не создавая дополнительного списка.

13. ЗАДАЧА «КОЛИЧЕСТВО СОВПАДАЮЩИХ ПАР»

Дан список чисел. Посчитайте, сколько в нем пар элементов, равных друг другу. Считается, что любые два элемента, равные друг другу образуют одну пару, которую необходимо посчитать

14. ЗАДАЧА «УНИКАЛЬНЫЕ ЭЛЕМЕНТЫ»

Дан список. Выведите те его элементы, которые встречаются в списке только один раз. Элементы нужно выводить в том порядке, в котором они встречаются в списке.

15. ЗАДАЧА «КЕГЕЛЬБАН»

N кеглей выставили в один ряд, занумеровав их слева направо числами от 1 до N . Затем по этому ряду бросили K шаров, при этом i -й шар сбил все кегли с номерами от l_i до r_i включительно. Определите, какие кегли остались стоять на месте.

Программа получает на вход количество кеглей N и количество бросков K . Далее идет K пар чисел l_i, r_i , при этом $1 \leq l_i \leq r_i \leq N$.

Программа должна вывести последовательность из N символов, где j -й символ есть “**I**”, если j -я кегля осталась стоять, или “**.**”, если j -я кегля была сбита.

16. ЗАДАЧА «ФЕРЗИ»

Известно, что на доске 8×8 можно расставить 8 ферзей так, чтобы они не били друг друга. Вам дана расстановка 8 ферзей на доске, определите, есть ли среди них пара бьющих друг друга.

Программа получает на вход восемь пар чисел, каждое число от 1 до 8 — координаты 8 ферзей. Если ферзи не бьют друг друга, выведите слово **NO**, иначе выведите **YES**.