



7500. Трёхбуквенная баллада

Решена

Лёгкая

Выкладывание поля для сапера оказалось неожиданно приятной задачей — ты словно медитировал, укладывая плитки на поле. И в процессе тебя осенило: пока Кодеруна нет, кто-то должен заботиться о цифровом мире. Почему бы не быть этим кем-то?

Ты благодаришь странного человечка с рюкзаком, спешишь дальше — и тут же видишь персонажа в беде.

На высоком подоконнике сидит придворный менестрель. Он сгорбился и утирает слёзы.

— Мое вдохновение — его как будто кто-то украл... — всхлипывает менестрель.

— Я должен сочинить балладу в честь Принцессы Брейс, но всё, что у меня выходит — это бессмысленные нагромождения букв и пафоса. Мне нужно нечто простое, ясное... Идеальное! Только палиндромы достойны её ушей!

— **Три буквы**, читающиеся одинаково вперёд и назад. Вот и всё, что я прошу... Только три... — он простирает тебе свиток с черновиком и умоляюще смотрит в глаза.

Формат ввода

В качестве аргументов вашей функции передаются 2 параметра:

- ballad* — текст баллады. Длина баллады не превышает $3 \cdot 10^5$. Гарантируется, что баллада может содержать только строчные буквы английского алфавита и пробелы. Также гарантируется, что баллада не пуста. Баллада не может начинаться с пробела или заканчиваться им.
- Натуральное число *n* — длина баллады.

Формат вывода

В качестве ответа ваша программа должна вернуть одно число *x* — количество способов вычеркнуть из баллады **все пробелы** и некоторые буквы таким образом, чтобы осталось **три буквы**, которые образуют палиндром. Палиндромом называется строка, которая читается одинаково слева направо и справа налево. Два способа вычеркивания считаются различными, если найдется хотя бы один индекс, такой, что в первом способе буква с таким индексом в предложении вычеркнута, а во втором — нет.

Ограничения

Ограничение времени	2 с
Ограничение памяти	256 МБ

Пример 1

Ввод	Вывод
treasure	8

Пример 2

Ввод	Вывод
you will never find the treasure	146

Теги

coderun boost challenge

```
1 import string
2 def solve(ballad: str, n: int) -> int:
3     alphabet_list = [0 for _ in range(26)]
4     prefix_symbols_sum=[]
5     for symbol in ballad:
6         if symbol==' ':
7             prefix_symbols_sum.append(alphabet_list.copy())
8             continue
9             alphabet_list[ ord(symbol) - ord('a')]+=1
10            prefix_symbols_sum.append(alphabet_list.copy())
11
12    count=0
13    for i in range(1,n-1):
14        if ballad[i]==' ':
15            continue
16
17        for symbol_ind in range(26):
18            add=prefix_symbols_sum[i-1][symbol_ind]*(prefix_symbols_sum[-1][symbol_ind]+1)
19            count+=add
20
21
22    return count
23
24
25
26
```