

# Яндекс. Тренировки по алгоритмам июнь 2021, занятие 4

## I. Контрольная по ударениям

Ограничение времени	1 секунда
Ограничение памяти	64Mb
Ввод	стандартный ввод или input.txt
Вывод	стандартный вывод или output.txt

Учительница задала Пете домашнее задание — в заданном тексте расставить ударения в словах, после чего поручила Васе проверить это домашнее задание. Вася очень плохо знаком с данной темой, поэтому он нашел словарь, в котором указано, как ставятся ударения в словах. К сожалению, в этом словаре присутствуют не все слова. Вася решил, что в словах, которых нет в словаре, он будет считать, что Петя поставил ударения правильно, если в этом слове Петей поставлено ровно одно ударение. Оказалось, что в некоторых словах ударение может быть поставлено больше, чем одним способом. Вася решил, что в этом случае если то, как Петя поставил ударение, соответствует одному из приведенных в словаре вариантов, он будет засчитывать это как правильную расстановку ударения, а если не соответствует, то как ошибку. Вам дан словарь, которым пользовался Вася и домашнее задание, сданное Петей. Ваша задача — определить количество ошибок, которое в этом задании насчитает Вася.

### Формат ввода

Вводится сначала число  $N$  — количество слов в словаре ( $0 \leq N \leq 20000$ ). Далее идет  $N$  строк со словами из словаря. Каждое слово состоит не более чем из 30 символов. Все слова состоят из маленьких и заглавных латинских букв. В каждом слове заглавная ровно одна буква — та, на которую попадает ударение. Слова в словаре расположены в алфавитном порядке. Если есть несколько возможностей расстановки ударения в одном и том же слове, то эти варианты в словаре идут в произвольном порядке. Далее идет упражнение, выполненное Петей. Упражнение представляет собой строку текста, суммарным объемом не более 300000 символов. Строка состоит из слов, которые разделяются между собой ровно одним пробелом. Длина каждого слова не превышает 30 символов. Все слова состоят из маленьких и заглавных латинских букв (заглавными обозначены те буквы, над которыми Петя поставил ударение). Петя мог по ошибке в каком-то слове поставить более одного ударения или не поставить ударения вовсе.

### Формат вывода

Выведите количество ошибок в Петинем тексте, которые найдет Вася.

#### Пример 1

Ввод

Вывод

```
4
cAnnot
cannOt
f0und
pAge
thE pAge cAnnot be found
```

```
2
```

#### Пример 2

Ввод

Вывод

```
4
cAnnot
cannOt
f0und
pAge
The PAGE cannot be found
```

```
4
```

# Пример 3

Ввод

0  
ciFqrIxe LgvqquN zvd1hnXJ tizFPXtv JxqWqgnR CabaJ hFYoqbhH UyfiTX0 YvAylvnc ymtHHfnqh bmTLsEnh hikroekt dtVSftFBz ofQrMfo jGTGofv dVR

## Примечания

Примечания к примерам тестов

1. В слове сAnnot, согласно словарю возможно два варианта расстановки ударения. Эти варианты в словаре могут быть перечислены в любом порядке (т.е. как сначала сAnnot, а потом сanнOt, так и наоборот). Две ошибки, совершенные Петей — это слова be (ударение вообще не поставлено) и fouNd (ударение поставлено неверно). Слово thE отсутствует в словаре, но поскольку в нем Петя поставил ровно одно ударение, признается верным. 2. Неверно расставлены ударения во всех словах, кроме The (оно отсутствует в словаре, в нем поставлено ровно одно ударение). В остальных словах либо ударные все буквы (в слове PAGE), либо не поставлено ни одного ударения.

Язык Python 3.9 (PyPy 7.3.11)

Набрать здесь Отправить файл

```
1 from collections import defaultdict
2
3 def main():
4     N = int(input().strip())
5
6     dct = defaultdict(list)
7     for _ in range(N):
8         word = input().strip()
9
10        for i in range(len(word)):
11            if word[i].isupper():
12                dct[word.lower()].append(i)
13
14    seq = input().split()
15    all_errors = 0
16
17    for word in seq:
18
19        n_upper = -1
20        error = False
21        for i in range(len(word)):
22            if word[i].isupper():
23                if n_upper == -1:
24                    n_upper = i
25                else:
26                    error = True
27                    break
28
29        if error or n_upper == -1:
30            all_errors += 1
31        else:
32            word_lower = word.lower()
33            if word_lower in dct.keys():
34                if n_upper not in dct[word_lower]:
35                    all_errors += 1
36
37
38    print(all_errors)
```

Отправить

Предыдущая

Следующая