



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Prawie anagramy – LOGIA 24 (2023/24), etap 3

Treść zadania

Anagram to napis powstały z innego napisu przez przestawienie liter. Anagramami są np. wyrazy *kot* i *kto*. Prawie anagramami będziemy nazywać napisy, które po dokonaniu dokładnie jednej poprawki będą anagramami. Poprawka może polegać na dodaniu, usunięciu lub zamianie jednej litery. Napisz program, który sprawdzi, czy dwa napisy są prawie anagramami i wypisze, jaką poprawkę należy wykonać w pierwszym napisie.

Wejście

Dwa wiersze, w każdym znajduje się niepusty napis złożony z małych liter alfabetu łacińskiego o długości nie większej niż 1000. Przyjmij, że napisy nie są anagramami.

Wyjście

Jeden wiersz zawierający:

- tekst **nie**, gdy napisy nie są prawie anagramami;
- tekst **tak brak x**, gdy napisy są prawie anagramami i należy dodać literę **x** do pierwszego napisu;
- tekst **tak nadmiar x**, gdy napisy są prawie anagramami i należy usunąć literę **x** z pierwszego napisu;
- tekst **tak x na y**, gdy napisy są prawie anagramami i należy zamienić literę **x** na **y** w pierwszym napisie.

Przyjmij, że między napisami jest pojedyncza spacja.

Przykłady:

Wejście	krab barka	barka krab	ola ela	mama ma
Wyjście	tak brak a	tak nadmiar a	tak o na e	nie

Omówienie rozwiązania

Jeśli długości napisów różnią się co najmniej o 2, to nie są one prawie anagramami. Dla pozostałych ciągów znaków rozwiązanie zadania sprowadza się do zliczania liter. Przy czym dla pierwszego napisu zwiększamy licznik odpowiadający danej literze, a dla drugiego zmniejszamy.

```
dla każdej litera w napis1
    ile[litera] ← ile[litera] + 1
dla każdej litera w napis2
    ile[litera] ← ile[litera] - 1
```

Przeglądając listę `ile` zliczamy wartości 0, 1 i -1.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
ile0 ← 0
ile1 ← 0
ilem1 ← 0
dla każdej liczba w ile
    jeżeli liczba = 0 to ile0 ← ile0 + 1
    jeżeli liczba = 1 to ile1 ← ile1 + 1
    jeżeli liczba = -1 to ilem1 ← ilem1 + 1
```

W przypadku prawie anagramów:

- napisy będą miały taką samą długość oraz lista *ile* będzie zawierała dwadzieścia cztery zera, jedną wartość 1 i jedną wartość -1 albo
- długości napisów różnią się o 1 oraz lista *ile* będzie zawierała dwadzieścia pięć zer oraz jedną wartość 1 albo jedną wartość -1.

W pierwszym przypadku należy wypisać komunikat **tak x na y**, gdzie **x** to litera odpowiadająca wartości 1, a **y** wartości -1. W drugim przypadku 1 oznacza nadmiarową literę, a -1 jej brak. Litery identyfikujemy po pozycji na liście *ile*.

Rozwiązanie w języku Python

```
1 def prawieAnagramy(napis1, napis2):
2     #ustalenie długości napisów
3     dl1 = len(napis1)
4     dl2 = len(napis2)
5     if abs(dl1 - dl2) >= 2:
6         return 'nie'
7     #zliczanie znaków dla potencjalnych prawie anagramów
8     ile = [0] * 26
9     for z in napis1:
10        ile[ord(z) - 97] += 1
11    for z in napis2:
12        ile[ord(z) - 97] -= 1
13    #zliczanie 0, 1, -1
14    ile0 = 0; ile1 = 0; ilem1 = 0
15    for i in ile:
16        if i == 0: ile0 += 1
17        if i == 1: ile1 += 1
18        if i == -1: ilem1 += 1
19
20    #dla równych długości
21    if dl1 == dl2:
22        if ile0 == 24 and ile1 == 1 and ilem1 == 1:
23            return 'tak ' + chr(ile.index(1) + 97) + ' na ' +
24                chr(ile.index(-1) + 97)
25        return 'nie'
26    #dla różnych długości
27    if ile0 != 25:
28        return 'nie'
29    if ile1 == 1:
30        return 'tak nadmiar '+chr(ile.index(1) + 97)
31    return 'tak brak '+chr(ile.index(-1) + 97)
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
32 a = input()
33 b = input()
34 print(prawieAnagramy(a, b))
```

W kodzie użyto:

- `ord` – wynikiem jest kod liczbowy znaku,
- `chr` – wynikiem jest znak odpowiadający liczbie,
- `index` – wynikiem jest indeks pierwszego wystąpienia poszukiwanego elementu.

Lista `ile` jest indeksowana od 0 do 25, gdzie literze `a` odpowiada indeks 0, a literze `z` indeks 25. Ustalając indeks należy od kodu liczbowego znaku odjąć kod liczbowy znaku `a`, który wynosi 97 (`ord(z) - 97`). Przy odtwarzaniu znaku odwołujemy się do jego kodu, który uzyskujemy dodając do indeksu kod liczbowy znaku `a` (`ile.index(...) + 97`).

Testy

Testy zostały tak dobrane, by sprawdzić wszystkie warianty opisane w treści zadania. Pierwsza grupa testów obejmowała napisy o różnej długości. W pozostałych grupach testów dodatkowo występowały pary napisów o równej długości, przy czym nie wszystkie były prawie anagramami.

Grupa testów	Test	Wynik
I	opeqa	tak nadmiar q
	aepo	
	dor	tak brak x
	xord	
II	zvxp	nie
	px	
	mole	tak o na i
	mile	
III	barok	tak k na n
	baron	
	logo	nie
	loggol	
IV	algorytmeuklidesaczylinw	tak brak d
	euklidesaalgorytmnwdczyli	
	stuletni	tak t na l
	intellus	
V	algorytmicznoprogramistyczny	nie
	prograistycznyalgorytmiznot	
	qazxswedcvfrtgbnhyujmkiolpsitoeratostenesaplokmnjuhbvgytfcxdreszawq	tak nadmiar a
	sitoeratostenesqqwweerrttyuuiioopllkkjjhggffddssaaazxcvbbnnmm	
VI	programowanivdynamiczne	tak v na d
	ddnamiczneprogramowaniy	
	abrakadabra	nie
	aakadabbayy	
VII	mnbvcxzasdfghjklpoiuytrewqqwertyuioplkjhgfdsazxcvbnmlcbypiersze	tak brak w
	lqqiwwceezrrbtttyyppuuiieoorppwaassszddeffgghjjkllmmnnbbvvccxxzz	

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

[illegible]