

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Nasłuch – LOGIA 22 (2021/22), etap 2

Treść zadania

Sławek pracuje w Centrum Astronomicznym Bitmir. Jego zadaniem jest sprawdzanie zgodności wzorów w sygnałach radiowych. Każdy sygnał reprezentowany jest przez ciąg liter. W żargonie trafienie „wow!” następuje wtedy, gdy w dwóch porównywanych sygnałach litera na tej samej pozycji okaże się taka sama. Jeżeli litera nie występuje na tej samej pozycji, ale występuje w obu sygnałach, to mamy trafienie „ooo!”. Wystąpienie litery może być wykorzystane tylko w jednym trafieniu, pierwszeństwo ma trafienie „wow!”. Na przykład w sygnałach "aaaaabbbb" i "addcccaa" występuje jedno trafienie „wow!” i dwa trafienia „ooo!”.

Napisz program, który wczyta ze standardowego wejścia dwa sygnały i wypisze wynik porównania zgodności sygnałów w postaci liczby całkowitej nieujemnej, której cyfra dziesiątek oznacza liczbę trafień „wow!”, a cyfra jedności liczbę trafień „ooo!”.

Wejście:

Dwa napisy złożone z małych liter alfabetu łacińskiego oddzielone spacją o długości od 1 do 9 każdy.

Wyjście:

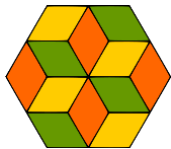
Liczba całkowita z przedziału $[0,90]$, której cyfra dziesiątek oznacza liczbę trafień „wow!”, a cyfra jedności liczbę trafień „ooo!”.

	Przykład 1	Przykład 2	Przykład 3
Wejście	aaabbbb bbbbaabbbb	abcc aacawxyz	aaabbbb aabb
Wyjście	6	20	31
Komentarz	Powtarzają się trzykrotnie litery „a” oraz „b”. Żadna litera nie występuje na tej samej pozycji.	Litery „a” oraz „c” znajdują się na tych samych pozycjach i tworzą trafienia „wow!”.	Dwie litery „a” i litera „b” tworzą trafienia „wow!”. Jedna litera „b” tworzy trafienie „ooo!”.

Omówienie rozwiązania

Zadanie jest uogólnieniem zadania „Nasłuch” z pierwszego etapu Konkursu Logia w roku szkolnym 2021/22. Porównywane słowa mogą być różnej długości, litery mogą się powtarzać. Techniczną zmianą jest zastąpienie 10-ciu cyfr przez 26 liter.

W przykładowym rozwiązaniu wykorzystamy pomocniczą tablicę, w której będziemy zaznaczać wystąpienia poszczególnych liter. Litery, które występują na tych samych pozycjach zwiększą wartość zmiennej trafienia o 10, ponadto litery z napisu U będą zwiększały o 1 odpowiadający im numer w tablicy zliczone, zaś litery z napisu W będą zmniejszały o 1 odpowiadający im numer (litera „a” ma numer 0, „b” - 1,..., „z” - 25). Zsumowanie wartości bezwzględnych w tej tablicy da liczbę niesparowanych liter. Odejmując tę wartość od sumy długości napisów i dzieląc przez dwa otrzymamy liczbę par w obu napisach. Należy odjąć pary z trafieniami „wow”, aby otrzymać liczbę trafień „ooo”.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie w postaci listy kroków:

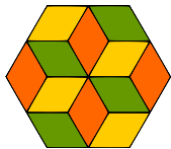
```
wczytaj U, W
trafienia ← 0
zainicjuj tablicę zliczone[0..25] zerami
maks_dlugosc ← maksimum(dlugosc(U), dlugosc(W))
dla każdego i z [0, 1, 2,..., maks_dlugosc - 1] wykonuj kroki 6-12
    jeżeli i < dlugosc(U) wykonaj kroki 7-11
        w przeciwnym razie wykonaj krok 12
            zwiększ o 1 wartość zliczone[numer i-tego znaku z U]
            jeżeli i < dlugosc(W) wykonaj kroki 9-11
                zmniejsz o 1 wartość zliczone[numer i-tego znaku z W]
                jeżeli znaki o numerach i są równe w U i W wykonaj
                    krok 11
                zwiększ o 10 wartość trafienia
            zmniejsz o 1 wartość zliczone[numer i-tego znaku z W]
suma ← 0
dla każdego j z [0, 1, 2,..., 25] wykonaj krok 15
    suma ← suma + wartość_bezwzględna(zliczone[j])
trafienia ← trafienia + (dlugosc(U) + dlugosc(W) - suma) div 2
                    - trafienia div 10
wyświetl trafienia
```

W pierwszych trzech liniach wczytujemy dwa napisy oraz ustawiamy początkowe wartości zmiennych na zero. W linii 4 obliczamy długość dłuższego napisu i zapisujemy w zmiennej - tyle razy będziemy iterować w liniach 6-12. W liniach tych zwiększana jest wartość zmiennej *trafienia*, gdy litery na takich samych pozycjach są równe. Ponadto w tablicy zmieniane są wartości wystąpień odpowiadające numerom liter. W liniach 13-15 obliczana jest suma wartości bezwzględnych z tabeli *zliczone*. Następnie modyfikowana jest zmienna *trafienia*, w której mieliśmy zliczone trafienia „wow”, a teraz dodajemy trafienia „ooo”.

Rozwiązanie w języku Python

Wykorzystując rozwiązanie w postaci listy kroków napisanie rozwiązania w Python mogłoby wyglądać następująco:

```
1 U, W = input().split()
2 trafienia = 0
3 zliczone = 26*[0]
4 maks_dlugosc = max(len(U), len(W))
5 for i in range(maks_dlugosc):
6     if i < len(U):
7         zliczone[ord(U[i]) - ord('a')] += 1
8     if i < len(W):
9         zliczone[ord(W[i]) - ord('a')] -= 1
10        if U[i] == W[i]:
11            trafienia += 10
12    else:
13        zliczone[ord(W[i]) - ord('a')] -= 1
14 suma = 0
15 for j in range(26):
16     suma += abs(zliczone[j])
17 trafienia += (len(U) + len(W) - suma) // 2 - trafienia // 10
18 print(trafienia)
```



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Do obliczenia numeru litery wykorzystaliśmy funkcję `ord(z)`, której wartością jest kod ASCII znaku `z`. Odejmując kod ASCII litery „a” otrzymamy numer litery.

Inne rozwiązanie, w którym wykorzystujemy operacje na listach składanych:

```
1 U, W = input().split()
2 trafienia = 9 * sum(1 for i in range(min(len(U), len(W))) if U[i]==W[i])
3 trafienia += sum(min(U.count(z), W.count(z)) for z in set(U) & set(W))
4 print(trafienia)
```

W linii 2 zliczamy trafienia „wow” lecz nie mnożymy ich przez 10, lecz 9, aby w linii 3 już ich nie odejmować, jak to zrobiliśmy w poprzednim programie w linii 17. Sumujemy funkcją `sum()` listę złożoną z tylu jedynek, ile liter jest równych w obu napisach. W linii 3 lista składana jest tworzona przez zliczanie liczby liter w obu napisach i wybrania wartości mniejszej do sumy. Należy zauważyć, że zliczamy w ten sposób także trafienia „wow”. Bierzemy pod uwagę tylko te znaki, które występują w obu napisach. Konstrukcja `set(U) & set(W)` zwraca część wspólną obu zbiorów, czyli litery, które występują w obu napisach `U` oraz `W`. Z kolei metoda `U.count(z)` pozwala na zliczenie liczby wystąpień znaku `z` w napisie `U`.

Testy

Testy zostały tak dobrane, by sprawdzić m.in. czy są zliczane litery występujące na początku lub końcu napisu, jak zachowuje się program, gdy litery powtarzają się.

Grupa testów	Test	Wynik
I	abc cba	12
	abcd uxyz	0
	abcde bxyze	11
II	fghj jf	2
	kmnoprs skmnoprs	7
	abzaedfa zaaaaaada	14
III	zzzzzzza azzzzzzz	72
	bbaaaaaa aabb	4
	ddddddddd edddddddd	80
IV	cbabc abcba	22
	kmn kmnkmnkmn	30
	abcdedcba edcbabcde	26
V	fg fhgjfgfgf	11
	rtyeertye yrerter	6
	wer twerwerte	3