

# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

## Zadanie Wagi napisów – LOGIA 21 (2020/21), etap 2

### Treść zadania

Tolek każdej literze w słowie przypisuje liczbę całkowitą nieujemną, którą nazywa wagą litery. Pierwsza i ostatnia litera słowa ma wagę **1**, druga i przedostatnia wagę **2**, trzecia od początku i końca wagę **3**, itd. Następnie wszystkim samogłoskom redukuje wagę o jeden, a spółgłoskom zwiększa wagę o jeden. Wagę słowa oblicza sumując wszystkie wagi liter. Napisz program, który ułoży słowa według ich wag od najmniejszej do największej. Jeśli dwa słowa mają taką samą wagę, to występują wtedy w kolejności alfabetycznej.

#### Wejście:

W pierwszym i jedynym wierszu znajduje się od 3 do 1000 słów oddzielonych pojedynczymi odstępami, o długości od 2 do 30, składających się z małych liter alfabetu łacińskiego.

#### Wyjście:

Jeden wiersz zawierający słowa uporządkowane według reguł opisanych w treści zadania. Słowa oddzielone są pojedynczymi odstępami.

	<i>Przykład 1</i>	<i>Przykład 2</i>	<i>Przykład 3</i>
Wejście	<b>ala ma kota</b>	<b>wykonaj zadanie waga słowa</b>	<b>aaa cc am la</b>
Wyjście	<b>ma ala kota</b>	<b>waga słowa zadanie wykonaj</b>	<b>aaa am la cc</b>

*Wyjaśnienie do przykładu 1:*

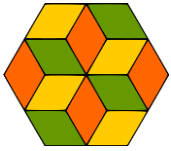
	<b>ala</b>	<b>ma</b>	<b>kota</b>
wagi na podstawie położenia	1+2+1	1+1	1+2+2+1
wagi po uwzględnieniu typu litery	0+3+0	2+0	2+1+3+0
waga słowa	3	2	6

### Omówienie rozwiązania

W zadaniu należało posortować dane według dwóch kryteriów. Pierwszym z nich była waga opisana w treści zadania, drugim kolejność alfabetyczna.

Waga słowa zależy od długości słowa oraz liczby samogłosek i spółgłosek w nim zawartych. Nie ma natomiast znaczenia położenie liter w ciągu znaków. Na przykład dla słów *kot* i *kto* waga wynosi 5.

Do obliczenia wagi na podstawie położenia możemy skorzystać ze wzoru na sumę ciągu arytmetycznego lub zastosować iterację. Niezależnie od wybranego sposobu należy rozpatrzyć dwa przypadki – dla słów parzystej długości oraz nieparzystej długości.



# Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Algorytm iteracyjny obliczający wagę bez uwzględniania typu liter:

1.  $suma \leftarrow 0$
2. dla każdego  $i$  od 1 do (długość słowa) // 2
3.  $suma \leftarrow suma + 2 * i$
4. jeśli (długość słowa) % 2 == 1
5.  $suma \leftarrow suma + (długość słowa) // 2 + 1$
6. wynik  $suma$

W wersji bez iteracji:

1.  $suma \leftarrow (1 + (długość słowa) // 2) * ((długość słowa) // 2)$
2. jeśli (długość słowa) % 2 == 1
3.  $suma \leftarrow suma + (długość słowa) // 2 + 1$
4. wynik  $suma$

Wystarczy jeszcze zliczyć liczbę samogłosek w słowie. Liczbę spółgłosek otrzymamy odejmując od długości słowa liczbę samogłosek. Algorytm zliczający samogłoski może wyglądać następująco:

1.  $l\_samoglosek \leftarrow 0$
2. dla każdego znaku w słowie
3. jeśli znak jest samogłoską, to
4.  $l\_samoglosek \leftarrow l\_samoglosek + 1$
5. wynik  $l\_samoglosek$

Waga uwzględniająca typ liter obliczona zostanie ze wzoru:

$$waga\ bez\ uwzględnienia\ typu\ liter - l\_samoglosek + (długość\ słowa - l\_samoglosek)$$

czyli

$$waga\ bez\ uwzględnienia\ typu\ liter + długość\ słowa - 2 * l\_samoglosek$$

Po wyznaczeniu wag możemy zaimplementować dowolny algorytm sortowania, np. bąbelkowy.

## Rozwiązanie w języku Python

Funkcja zliczająca samogłoski:

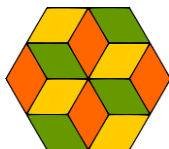
```
1. def ile(slowo):
2.     l_samoglosek = 0
3.     for zn in slowo:
4.         if zn in "aeiouy":
5.             l_samoglosek += 1
6.     return l_samoglosek
```

Można ją zastąpić listą składaną:

```
1. sum([1 for zn in slowo if zn in "aeiouy"])
```

Funkcja obliczająca wagę słowa:

```
1. def waga(slowo):
2.     ile = sum([1 for zn in slowo if zn in "aeiouy"])
3.     suma = (1 + len(slowo) // 2) * (len(slowo) // 2)
4.     if len(slowo) % 2 == 1:
5.         suma = suma + len(slowo) // 2 + 1
6.     return suma + len(slowo) - 2 * ile
```



## Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Rozwiązanie zadania w oparciu o sortowanie bąbelkowe:

```
1. def sortuj(lista):
2.     for i in range(0, len(lista) - 1):
3.         for j in range(len(lista) - 1 - i):
4.             if waga(lista[j]) > waga(lista[j + 1]):
5.                 lista[j], lista[j + 1] = lista[j + 1], lista[j]
6.             elif waga(lista[j]) == waga(lista[j + 1]) and lista[j] > lista[j + 1]:
7.                 lista[j], lista[j + 1] = lista[j + 1], lista[j]
8.     return lista
9.
10. dane = input().split()
11. wynik = sortuj(dane)
12. for ele in wynik:
13.     print(ele, end=" ")
```

### Testy

Testy zostały podzielone na 5 grup. Pierwsze trzy grupy zawierały słowa o różnych wagach, a dwie ostatnie słowa o powtarzających się wagach w celu sprawdzenia poprawności rozwiązania (uwzględnienie dwóch kryteriów sortowania). Ponadto testy uwzględniały ciągi znaków różnej długości oraz różną liczbę słów.

Grupa testów	Różne wagi	Długość słowa	Liczba słów
I	tak	od 2 do 4	3
II	tak	od 4 do 9	od 8 do 10
III	tak	od 2 do 15	50
IV	nie	wszystkie słowa tej samej długości	500
V	nie	od 2 do 30	1000