

Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

Zadanie Kodowanie – LOGIA 23 (2022/23), etap 2

Treść zadania

Ignacy koduje wiadomość. W tym celu wymyślił własną metodę. Polega ona na wypisaniu najpierw wszystkich samogłosek występujących w wiadomości w kolejności alfabetycznej, następnie spółgłosek też w kolejności alfabetycznej. Jeśli jakaś litera występuje więcej niż jeden raz, to Ignacy wypisuje ją raz i po niej liczbę jej wystąpień. Napisz program kodujący wiadomość zgodnie z metodą Ignacego.

Wejście

Liczba całkowita **n** z zakresu od 1 do 100 określająca liczbę wierszy kodowanej wiadomości.

W kolejnych **n** wierszach znajduje się tekst składający się z małych liter alfabetu łacińskiego (samogłoski alfabetu łacińskiego to: a, e, i, o, u, y). Każdy wiersz tekstu składa się z co najmniej jednej litery i jest nie dłuższy niż 100 liter.

Wyjście

n wierszy tekstu – zakodowana wiadomość

Przykłady:

Wejście	2 alabama kwiatusek	1 ogrodzenie	3 akwarela butonierka zryw
Wyjście	a4blm aeiuk2stzw	e2io2dgnrz	a3eklrw aeioubknrt yrwz

Omówienie rozwiązania

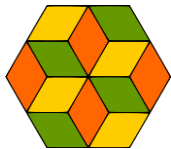
Zacznijmy od alfabetu łacińskiego – składa się on z 26 liter, które są na klawiaturze komputera i nie trzeba znać kolejności liter alfabetu na pamięć. Wystarczy go wygenerować wykorzystując funkcję `chr(kod)`, zamieniającą kod ASCII na odpowiednią literę alfabetu.

```
1 alfabet = ''
2 for i in range(26):
3     alfabet += chr(97 + i)
4 print(alfabet)
```

Po uruchomieniu tego programu zostanie wypisany alfabet złożony z małych liter (kody ASCII małych liter rozpoczynają się od 97).

abcdefghijklmnopqrstuvwxyz

Zanim zaczniemy rozwiązywać zadanie omówimy parametry funkcji `print()`. Zazwyczaj podaje się tylko dane do wypisania na ekranie – tekst, liczbę, listę itp. Funkcja `print()` posiada jeszcze dwa parametry opcjonalne: `sep = 'wartość'` oraz `end = 'wartość'`. Parametr `sep` powoduje, że



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

podczas wypisywania kilku wartości rozdzielonych przecinkiem, domyślny separator, jakim jest spacja, zostanie zamieniony na podany. Na przykład:

```
print('Ala', 'ma', 'kota', sep = '#')
```

zostanie wypisany napis **Ala#ma#kota**

Parametr `end` spowoduje zmianę znaku końca wiersza na podany, czyli po wypisaniu wartości kursor nie przejdzie do następnej linii, tylko zostanie wypisana podana wartość.

Na przykład polecenie:

```
print('Ala', 'ma', 'kota', sep = '#', end = '$')  
print('drugi')
```

wypisze na ekranie trzy napisy oddzielone znakiem '#' zamiast spacji, a kursor nie zostanie przeniesiony do następnej linii, ponieważ zostanie zastąpiony znakiem '\$'. Na ekranie pojawi się napis:

Ala#ma#kota\$drugi

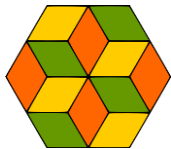
Metodę zliczania występujących samogłosek i spółgłosek można wybrać, sugerując się oczekiwanym wynikiem. Należy wypisywać w kolejności alfabetycznej najpierw samogłoski, potem spółgłoski. Dlatego warto zliczać kolejno samogłoski i wypisywać je, jeśli dana samogłoska wystąpiła w badanym słowie, oraz liczbę jej wystąpień, jeśli jest większa niż jeden. Podczas wypisywania separator zastępujemy pustym znakiem ' ' (2 apostrofy) i tak samo robimy ze znakiem końca wiersza.

```
1 sam = 'aeiouy'  
2 w = input()  
3 for j in range(len(sam)):  
4     ile = w.count(sam[j])  
5     if ile == 1:  
6         print(sam[j], end='')  
7     elif ile > 1:  
8         print(sam[j], ile, sep = ' ', end = ' ')
```

Na przykład dla wczytanego napisu **aeaeaeooouyo** zostanie wypisany wynik **a3e3o4uy**.

Zliczanie spółgłosek jest identyczne, tylko wartość zmiennej pomocniczej, to wszystkie spółgłoski w kolejności alfabetycznej.

Program na początku wczytuje liczbę całkowitą **n** określającą liczbę wierszy kodowania. Potem są zadeklarowane zmienne przechowujące samogłoski i spółgłoski w kolejności alfabetycznej. W pętli `for` powtarzamy **n** razy wczytywanie tekstu. Dla każdego tekstu zliczamy kolejno samogłoski, jeśli dana samogłoska występuje raz, to wypisujemy ją, jeśli więcej razy, to wypisujemy ją i liczbę jej wystąpień. Po samogłoskach zliczamy spółgłoski. Cały czas wszystkie wartości wypisujemy znak po znaku, pusty wiersz `print()` jest wstawiony dopiero po przeanalizowaniu całego wczytanego napisu.



Przedmiotowy Konkurs Informatyczny LOGIA powołany przez Mazowieckiego Kuratora Oświaty

```
1 n = int(input())
2 sam = 'aeiouy'
3 sp = 'bcdfghjklmnpqrstvwxyz'
4 for i in range(n):
5     w = input()
6     for j in range(len(sam)):
7         ile = w.count(sam[j])
8         if ile == 1:
9             print(sam[j], end='')
10        elif ile > 1:
11            print(sam[j], ile, sep = '', end = '')
12    for j in range(len(sp)):
13        ile = w.count(sp[j])
14        if ile == 1:
15            print(sp[j], end='')
16        elif ile > 1:
17            print(sp[j], ile, sep = '', end = '')
18    print()
```

Testy

Testujemy działanie programu dla przykładów z treści zadania. Warto również samemu wymyśleć i uruchomić testy zawierające dłuższe teksty.

Na przykład dla testu:

```
3
elementarz
serdeczniejimilej
abrakadabrahokuspokus
```

wynikiem będzie:

```
ae3lmnrtz
e4i3cdj2lmnrsz
a5o2u2b2dhk3pr2s2
```

Dla testu:

```
2
mhkajircduodjffichvnsgpzheessakrwrpbmjemrfvbjdqzbu
oovigzxesglopeufwdhzcpcnaco zghehxujxlygcszlelgt dvxzkdszgx
```

wynik będzie następujący:

```
a2e3i2ou2b3c2d3f3gh3j4k2m3np2qr4s3v2wz2
ae4io4u2yc3d3fg6h3jk14np2s3tv2wx5z6
```