Zadanie Naj… – LOGIA 24 (2023/24), etap 2

Treść zadania

Adam analizuje napisy złożone z małych liter alfabetu polskiego (32 litery):

a	ą	b	c	ć	d	e	ę	f	g	h	i	j	k	l	ł	m	n	ń	o	ó	p	r	s	ś	t	u	w	y	z	ź	ż

Z każdego napisu wybiera literę najwcześniejszą w alfabecie i najpóźniejszą.

Pomóż Adamowi i napisz program, który wczyta napis, znajdzie i wypisze najwcześniejszą oraz najpóźniejszą literę występujące w napisie, a także odległość między nimi w alfabecie.

Wejście

Niepusty napis złożony z małych liter alfabetu polskiego o długości nie większej niż 1000 znaków.

Wyjście

Dwie litery i liczba całkowita nieujemna oddzielone spacjami.

Przykłady:

Wejście	**abrakadabra**	**żółw**	**misiek**
Wyjście	**a r 22**	**ł ż 16**	**e s 17**

Omówienie rozwiązania

Zadanie polega na analizie napisu składającego się z małych liter alfabetu polskiego. Z napisu należy wybrać literę najwcześniejszą i najpóźniejszą w alfabecie, a następnie obliczyć odległość w alfabecie między nimi. Program powinien wczytać napis i wypisać trzy dane – dwie litery, najwcześniejszą oraz najpóźniejszą występującą w napisie oraz liczbę – odległość między nimi w alfabecie.

Ponieważ alfabet zawiera również polskie znaki diakrytyczne, nie można skorzystać z kodów ASCII do przyporządkowania literom alfabetu kolejnych liczb, należy wybrać inne rozwiązanie. Gdy utworzymy napis, który zawiera litery w kolejności alfabetycznej, to pozycja litery w napisie będzie pozycją litery w alfabecie. W ten sposób możemy wczytany napis zamienić na listę liczb. Z listy wybieramy liczbę najmniejszą i największą, a następnie znajdujemy odpowiadające im litery. Ponadto liczymy odległość jako różnicę liczby największej i najmniejszej.

Rozwiązanie w języku Python

Rozwiązanie korzysta z mechanizmu list składanych (ang. *list comprehensions*), który umożliwia szybkie i efektywne tworzenie nowych list poprzez przekształcenie elementów danej sekwencji. W tym przypadku, konstrukcja listy składanej `[alfabet.index(x) for x in napis]` jest używana do iteracji po znakach napisu. Dla każdego znaku `x` w zmiennej `napis`, wykonywane jest wyrażenie `alfabet.index(x)`, które szuka pozycji tego znaku w alfabecie. Działanie to przekształca każdy znak napisu na odpowiadający mu indeks w zmiennej `alfabet`, co pozwala na reprezentację napisu jako listy liczb. Ta lista indeksów jest następnie wykorzystywana do dalszych operacji, takich jak znalezienie minimalnej i maksymalnej wartości, co odpowiada odpowiednio najwcześniejszej i najpóźniejszej literze w alfabecie spośród liter występujących w napisie. Do znalezienia wartości minimalnej i maksymalnej wykorzystujemy funkcje `min()` i `max()`.

```
 1  def znajdz(napis):
 2      alfabet = "aąbcćdeęfghijklłmnńoóprsśtuwyzźż"
 3
 4      lista = [alfabet.index(x) for x in napis]
 5      p = min(lista)
 6      k = max(lista)
 7
 8      return alfabet[p], alfabet[k], k - p
 9
10  napis = input()
11  wynik = znajdz(napis)
12  print(wynik[0], wynik[1], wynik[2])
```

Testy

Najpierw testujemy zadanie na przykładach z treści zadania. Potem warto sprawdzić proste, krótkie słowa, które nie zawierają polskich znaków diakrytycznych. W testach nie powinno zabraknąć sprawdzenia, jak algorytm radzi sobie z dłuższym napisem i czy poprawnie identyfikuje skrajne litery zarówno z alfabetu łacińskiego jak i polskiego. Ważnym testem jest również napis zawierający wszystkie litery alfabetu.

Test	**Wynik**
kij	i k 2
glif	f l 6
logia	a o 19
algorytmika	a y 28
www	w w 0
"mn" * 100	m n 1
wół	ł w 12
nietoperz	e z 23
czwórki	c z 26
ćąlwaęócktosjhińbpugśfżźrzełmynd	a ż 31