

Mutual love

The word LOVE exhibits a perfect symmetry: the letters are evenly positioned compared to the middle of the alphabet. If this condition is met, then we say that the love between the letters of the word is mutual.



The love is mutual between the letters of the word LOVE.

Assignment

Determine if the love between the letters of a word is mutual. To do this, just follow these steps:

- Write a function `alphabetPositions` to which a word must be passed as a string argument. The function must return a tuple with the positions in the alphabet of each letter in the word. Here the letter A is at position zero in the alphabet, the letter B is on position one, The function cannot distinguish between uppercase and lowercase letters, and all characters of the given word that are not a letter, should be ignored.
- Write a function `mutual` to which a sequence (a list or tuple) of integers and an even natural number $n \in \mathbb{N}_{>0}$ must be passed. For all numbers i in the given set it must be that $0 \leq i < n$. The function is required to return a Boolean value indicating whether the numbers in the given set can be grouped per two, so that the numbers of each pair are symmetrically positioned with respect to the center of the series of numbers $0, 1, 2, \dots, n - 1$.
- Use the functions `alphabetPositions` and `mutual` to write a function `mutualLove` to which a word must be passed as a string argument. The function must return a Boolean value indicating whether or not the love is mutual between the letters of the word. The function cannot make a distinction between uppercase and lowercase letters, and all characters of the given word that are not a letter, should be ignored.

Example

```
>>> alphabetPositions('LOVE')
(11, 14, 21, 4)
>>> alphabetPositions('mutual')
(12, 20, 19, 20, 0, 11)

>>> mutual((11, 14, 21, 4), 26)
True
>>> mutual([12, 20, 19, 20, 0, 11], 26)
False

>>> mutualLove('LOVE')
True
>>> mutualLove('mutual')
False
```

Het woord LOVE vertoont een perfecte symmetrie: de letters zijn immers gelijkmatig

gepositioneerd ten opzichte van het midden van het alfabet. Indien aan deze voorwaarde voldaan is, dan zeggen we dat de liefde tussen de letters van het woord wederzijds is.

ABCDEF|GHIJKLMN|OPQRSTUVWXYZ

De liefde is wederzijds tussen de letters van het woord LOVE.

Opgave

Bepaal of de liefde tussen de letters van een gegeven woord wederzijds is. Hiervoor ga je als volgt te werk:

- Schrijf een functie `alfabetposities` waaraan een woord als een stringargument moet doorgegeven worden. De functie moet een tuple teruggeven met de posities in het alfabet van elke letter in het woord. Hierbij staat de letter A op positie nul in het alfabet, de letter B op positie één, De functie mag geen onderscheid maken tussen hoofdletters en kleine letters, en alle karakters van het gegeven woord die geen letter zijn, moeten genegeerd worden.
- Schrijf een functie `wederzijds` waaraan een reeks (een lijst of een tuple) van natuurlijke getallen en een even natuurlijke getal $n \in \mathbb{N}_0$ moeten doorgegeven worden. Voor alle getallen i in de gegeven reeks moet gelden dat $0 \leq i < n$. De functie moet een Booleaanse waarde teruggeven, die aangeeft of de getallen in de gegeven reeks per twee kunnen gegroepeerd worden, zodat de getallen van elk paar symmetrisch gepositioneerd zijn ten opzichte van het midden van de getallenreeks $0, 1, 2, \dots, n - 1$.
- Gebruik de functies `alfabetposities` en `wederzijds` om een functie `wederzijdseLiefde` te schrijven waaraan een woord als een stringargument moet doorgegeven worden. De functie moet een Booleaanse waarde teruggeven, die aangeeft of de liefde tussen de letters van het woord al dan niet wederzijds is. De functie mag hierbij geen onderscheid maken tussen hoofdletters en kleine letters, en alle karakters van het gegeven woord die geen letter zijn, moeten genegeerd worden.

Voorbeeld

```
>>> alfabetposities('LOVE')
```

```
(11, 14, 21, 4)
```

```
>>> alfabetposities('mutual')
```

```
(12, 20, 19, 20, 0, 11)
```

```
>>> wederzijds((11, 14, 21, 4), 26)
```

```
True
```

```
>>> wederzijds([12, 20, 19, 20, 0, 11], 26)
```

```
False
```

```
>>> wederzijdseLiefde('LOVE')
```

```
True
```

```
>>> wederzijdseLiefde('mutual')
```

```
False
```