

Rotated words

Rotation is a weak form of encryption that consists of rotating every letter of a word. Rotating a letter is as much as shifting that letter over a set number of positions $r \in \mathbb{N}$ in the alphabet. If $r > 0$, this is a rotation to the right, and for $r < 0$ this is a rotation to the left. Here, the alphabet is seen as a circular list, which comes down to the fact that after the letter z , the letter a follows (when rotating to the right) and that the letter a is thus preceded by the letter z (when rotating to the left).

The rotation of the word `monty` over three positions gives the word `prqwb`. Here you can see for example that three positions of the letter `m` in the alphabet, is the letter `p`, and that by seeing this rotation as a circular movement, the letter `b` is three position right of the letter `y`. Two words are called *rotated words* if a r -rotation ($r \in \mathbb{N}$) consists that converts the first word in the second word. This way, `cobra` and `freud` are examples of rotated words, because a 3-rotation of the word `cobra` results in the word `freud`. Note that also a (-3)-rotation or a 23-rotation of the word `freud` results in the word `cobra`. The property of rotated words is, in other words, symmetrical.

Assignment

For this assignment, we only work with words that consist solely of lowercase letters. Your assignment is to find the rotated words in a list of words. To do so, you work as follows:

- Write a function `rotation` that prints the r -rotation ($r \in \mathbb{N}$) of a given word as a result. To this function two parameters must be given: the given word (a string that consist of lowercase letters) and the value r .
- The word stem of a word is the rotated word that starts with a given begin letter. Use the function `rotation` to write a function `wordstem` that prints the word stem of a given word that starts with a given begin letter. As a first obligatory parameter `word` a word should be given to the function that consists solely of lowercase letters. The function also has a second, optional parameter `beginletter`, which can be used to give the begin letter of the stem word. The letter `a` is used as a standard begin letter.
- Use the function `wordstem` to write a function `rotatedWords`. To this function a list of words (consisting of lowercase letters) must be given. The function must print a collection of tuples as a result, where every tuple contains all words from the given word list that are rotated words of each other. Every tuple from the collection contains two or more words, that must be ordered alphabetically.

Example

```
>>> rotation('hal', 1)
'ibm'
>>> rotation('monty', 3)
'prqwb'
>>> rotation('python', 33)
'wfaovu'
>>> rotation('shrubbery', -3)
'peoryybov'

>>> wordstem('hal')
'ate'
```

```

>>> wordstem(word='monty', beginletter='x')
'xzyej'
>>> wordstem('python')
'ajeszy'
>>> wordstem('shrubbery', beginletter='f')
'fuehoorel'

>>> rotatedWords(['bubba', 'primero', 'fyffe', 'pippo', 'sulphur'])
{('primero', 'sulphur'), ('bubba', 'fyffe', 'pippo')}
>>> rotatedWords(['beowulf', 'harahan', 'leveler', 'rhesian'])
{('harahan', 'leveler')}

```

Rotatie is een zwakke vorm van encryptie die erin bestaat om elke letter van een woord te 'roteren'. Een letter roteren is zoveel als het verschuiven van de letter over een vast aantal posities $r \in \mathbb{N}$ in het alfabet. Wanneer $r > 0$ dan spreekt men van een rotatie naar rechts, en voor $r < 0$ spreekt men van een rotatie naar links. Hierbij wordt het alfabet als een circulaire lijst beschouwd, wat erop neerkomt dat na de letter z terug de letter a volgt (bij schuiven naar rechts), en dat de letter a dus voorafgegaan wordt door de letter z (bij schuiven naar links).

De rotatie van het woord `monty` over drie posities levert het woord `prqwb`. Hierbij zie je bijvoorbeeld dat drie posities rechts van de letter `m` in het alfabet de letter `p` staat, en dat door het circulair beschouwen van de lijst de letter `b` drie posities rechts van de letter `y` staat. Twee woorden worden *geroteerde woorden* genoemd indien er een r -rotatie ($r \in \mathbb{N}$) bestaat die het eerste woord omzet in het tweede woord. Zo zijn `cobra` en `freud` voorbeelden van geroteerde woorden, omdat een 3-rotatie van het woord `cobra` het woord `freud` oplevert. Merk op dat ook een (-3)-rotatie of een 23-rotatie van het woord `freud` het woord `cobra` oplevert. De eigenschap van de geroteerde woorden is met andere woorden symmetrisch.

Opgave

Voor deze opdracht werken we enkel met woorden die uitsluitend uit kleine letters bestaan. Je opdracht bestaat erin om uit een lijst van woorden alle geroteerde woorden te vissen. Hiervoor ga je als volgt te werk:

- Schrijf een functie `rotatie` die de r -rotatie ($r \in \mathbb{N}$) van een gegeven woord als resultaat teruggeeft. Aan deze functie moeten twee parameters doorgegeven worden: het gegeven woord (een string die enkel bestaat uit kleine letters) en de waarde r .
- De woordstam van een woord is het geroteerde woord dat start met een opgegeven beginletter. Gebruik de functie `rotatie` om een functie `woordstam` te schrijven die de woordstam van een gegeven woord teruggeeft die start met een opgegeven beginletter. Als eerste verplichte parameter `woord` moet aan de functie een woord doorgegeven worden dat enkel bestaat uit kleine letters. De functie heeft ook nog een tweede optionele parameter `beginletter`, waarmee de beginletter van het stamwoord kan opgegeven worden. Standaard wordt de letter `a` als beginletter gebruikt.
- Gebruik de functie `woordstam` om een functie `geroteerdeWoorden` te schrijven. Aan deze functie moet een lijst van woorden (bestaande uit kleine letters) doorgegeven worden. De functie moet als resultaat een verzameling van tuples teruggeven, waarbij elk tuple alle woorden uit de gegeven woordenlijst bevat die geroteerde woorden van elkaar zijn. Elk tuple uit de verzameling bevat dus twee of meer woorden, die alfabetisch gerangschikt moeten worden.

Voorbeeld

```
>>> rotatie('hal', 1)
'ibm'
>>> rotatie('monty', 3)
'prqwb'
>>> rotatie('python', 33)
'wfaovu'
>>> rotatie('shrubbery', -3)
'peoryybov'

>>> woordstam('hal')
'ate'
>>> woordstam(woord='monty', beginletter='x')
'xzyej'
>>> woordstam('python')
'ajeszy'
>>> woordstam('shrubbery', beginletter='f')
'fuehoorel'

>>> geroteerdeWoorden(['bubba', 'primero', 'fyffe', 'pippo', 'sulphur'])
{('primero', 'sulphur'), ('bubba', 'fyffe', 'pippo')}
>>> geroteerdeWoorden(['beowulf', 'harahan', 'leveler', 'rhesian'])
{('harahan', 'leveler')}
```