

Chemical words

In the periodic table each chemical element is assigned a symbolic name consisting of one, two or three letters. If a particular word can be written as a sequence of those symbolic names for chemical elements, then we say that this word is a *chemical word*. For example, catalysis is a chemical word, for it can be written as C-At-Al-Y-Si-S, the sequence of symbolic names for the elements carbon, astatine, aluminum, yttrium, silicon and sulphur. However, catalyst is not a chemical word. Below are a few examples of chemical word.

scientific terms	chemical word
basic	Ba-Si-C
halogens	H-Al-O-Ge-N-S
catalysis	C-At-Al-Y-Si-S
interferon	In-Te-Rf-E-r-O-N
psychogenesis	P-S-Y-C-Ho-Ge-Ne-Si-S
sepsis	Se-P-Si-S
syphilis	S-Y-P-H-I-Li-S

Assignment

Given is a text file [periodic_system.txt](#), that contains a list with information about the elements from the periodic table. Each line — except the first which is the header — contains the following information about an element: *i*) atom number, *ii*) symbolic representation, *iii*) English name, *iv*) Dutch name, and *v*) atom mass. The information fields are separated by a tab. The assignment consists of determining whether a given word is a chemical word or not. To do this, follow these steps:

1. Write a function `readSymbols` that takes the location of a text file as its argument. The content of this text file must be formatted as the file [periodic_system.txt](#). The function must return the list of symbolic names that occur in the second column of the given text file.
2. Write a function `longestPrefix` to which two parameters must be passed. The first parameter `word` is a string, and the second parameter `symbols` is a list of strings. The function should return the longest prefix of `word` that is also a symbolic name that occurs in the list `symbols`. For example, the symbolic names B and Be are both prefixes of the word beach. The function should, however, return the longest prefix, so in this case that would be the string Be. If none of the symbolic names from the list `symbols` is a prefix of the given word, then the function should return an empty string. When determining whether a symbolic name is a prefix of a word, capitals should not be taken into account. However, maintain the spelling of the symbolic names when returning the prefix: Be and not BE or be.

Terminology: A *prefix* consists of one or more letters in front of the word. For example, chem is a prefix of the word chemistry.

3. Write a function `chemicalWord` to which two parameters must be passed. The first parameter `word` is a string, and the second parameter `symbols` is a list of strings. If the word is a chemical word, then the function should return the word as a sequence of symbolic names for chemical elements. The symbolic names should be separated by a hyphen (-). If the word is not a chemical word, then the function should return an empty string.

The procedure for determining whether a word is a chemical word or not, and at the same

time to generate the notation as a succession of symbolic names, is illustrated in the table below, on the basis of the word *catalysis*. As long as the word is not reduced to the empty string, you determine the longest prefix of the word that is also a symbolic name for an element (obviously using the `longestPrefix` function). If a prefix is found that is not an empty string, then remove this prefix in front of the word and add it to the back the sequence of symbolic names (taking into account the placement of hyphens). Once the longest prefix yields an empty string, you may conclude that it will not be a chemical word, and so let the function return an empty string. However, if you can continue to repeat the above procedure until the original word has been reduced to the empty string, you may conclude that it will be a chemical word, and you have also found the notation as a sequence of symbolic names.

	word	longest prefix	chemical word
	catalysis	C	
	atalysis	At	C
	alysis	Al	C-At
	ysis	Y	C-At-Al
	sis	Si	C-At-Al-Y
	s	S	C-At-Al-Y-Si
			C-At-Al-Y-Si-S

Example

In the following interactive Python session we assume that the text file [periodic_system.txt](#) is located in the current directory.

```
>>> symbols = readSymbols('periodic_system.txt')
```

```
>>> longestPrefix('ages', symbols)
```

```
'Ag'
```

```
>>> longestPrefix('shop', symbols)
```

```
'S'
```

```
>>> longestPrefix('density', symbols)
```

```
"
```

```
>>> chemicalWord('catalysis', symbols)
```

```
'C-At-Al-Y-Sis'
```

```
>>> chemicalWord('basic', symbols)
```

```
'Ba-Si-C'
```

```
>>> chemicalWord('leisure', symbols)
```

```
"
```

In het periodiek systeem krijgt elk scheikundig element een symbolische naam toegekend die bestaat uit één, twee of drie letters. Als een bepaald woord kan geschreven worden als opeenvolging van die symbolische namen voor scheikundige elementen, dan zeggen we dat dit woord een *chemisch woord* is. Zo is bijvoorbeeld *katalyse* een chemisch woord. Het kan immers geschreven worden als K-At-Al-Y-Se, de opeenvolging van symbolische namen voor de elementen kalium, astaat, aluminium, yttrium en seleen. Het woord *katalysator* is daarentegen geen chemisch woord. Hieronder staan alvast enkele voorbeelden van chemische woorden weergegeven.

scheikundige term chemisch woord

basisch

Ba-Si-Sc-H

halogenen	H-Al-O-Ge-Ne-N
katalyse	K-At-Al-Y-Se
parenteraal	Pa-Re-N-Te-Ra-Al
psychogene	P-S-Y-C-Ho-Ge-Ne
sepsis	Se-P-Si-S
sereus	Se-Re-U-S

Opgave

Gegeven is een tekstbestand [periodiek_systeem.txt](#), dat een lijst met informatie over de elementen uit het periodiek systeem bevat. Elke regel — behalve de eerste die als hoofding geldt — bevat de volgende informatie over een element: *i)* atoomnummer, *ii)* symbolische voorstelling, *iii)* Engelstalige naam, *iv)* Nederlandstalige naam, en *v)* atoommassa. De informatievelden worden van elkaar gescheiden door een tab. Je opdracht voor deze opgave bestaat er in om voor een gegeven woord te bepalen of het een chemisch woord is of niet. Hiervoor ga je als volgt te werk:

1. Schrijf een functie `leesSymbolen` waaraan de locatie van een tekstbestand moet doorgegeven worden. De inhoud van dit tekstbestand moet opgemaakt zijn zoals het tekstbestand [periodiek_systeem.txt](#). De functie moet de lijst van symbolische namen teruggeven die voorkomen in de tweede kolom van het gegeven bestand.
2. Schrijf een functie `langstePrefix` waaraan twee parameters moeten doorgegeven worden. De eerste parameter `woord` is een string, en de tweede parameter `symbolen` is een lijst van strings. De functie moet de langste prefix van `woord` teruggeven die ook een symbolische naam is die voorkomt in de lijst `symbolen`. Zo zijn de symbolische namen `B` en `Be` beide prefixen van het woord `berg`. De functie moet echter de langste prefix teruggeven, dus gaat in dit geval de voorkeur naar de string `Be`. Indien geen enkele symbolische naam uit de lijst `symbolen` een prefix is van het gegeven woord, dan moet de functie een lege string als resultaat teruggeven. Bij het bepalen of een symbolische naam een prefix is van het gegeven woord, moet geen rekening gehouden worden met het verschil tussen hoofdletters en kleine letters. Behoud echter de schrijfwijze van de symbolische namen bij het teruggeven van de prefix: `Be` en niet `BE` of `be`.

Terminologie: Een voorvoegsel bestaande uit één of meer letters vooraan een woord noemen we een *prefix* van dat woord. Zo is bijvoorbeeld `chem` een prefix van het woord `chemie`.

3. Schrijf een functie `chemischWoord` waaraan twee parameters moeten doorgegeven worden. De eerste parameter `woord` is een string, en de tweede parameter `symbolen` is een lijst van strings. Indien het woord een chemisch woord is, dan moet de functie het woord als opeenvolging van symbolische namen voor scheikundige elementen teruggeven. Hierbij moeten de symbolische namen van elkaar gescheiden worden door een koppelteken (`-`). Indien het woord geen chemisch woord is, dan moet de functie een lege string teruggeven. De te volgen procedure om te bepalen of een woord een chemisch woord is of niet, en tegelijkertijd ook de schrijfwijze als opeenvolging van symbolische namen te genereren, wordt in onderstaande tabel geïllustreerd aan de hand van het woord `katalyse`. Zolang het woord niet gereduceerd is tot de lege string, bepaal je de langste prefix van het woord die ook een symbolische naam voor een element is (uiteeraard door gebruik te maken van de functie `langstePrefix`). Indien er een prefix wordt gevonden die geen lege string is, dan verwijder je deze prefix vooraan het woord en voeg je die achteraan toe aan de opeenvolging van symbolische namen (rekening houdend met het plaatsen van

koppeltekens). Van zodra de langste prefix een lege string oplevert, kan je besluiten dat het woord geen chemisch woord is en dus een lege string laten teruggeven door de functie. Indien je bovenstaande procedure echter kunt blijven herhalen tot het originele woord gereduceerd is tot de lege string, dan mag je besluiten dat het woord een chemisch woord is en heb je meteen ook de schrijfwijze als opeenvolging van symbolische namen gevonden.

woord	langste prefix	chemisch woord
katalyse	K	
atalyse	At	K
alyse	Al	K-At
yse	Y	K-At-Al
se	Se	K-At-Al-Y
		K-At-Al-Y-Se

Voorbeeld

Bij de onderstaande interactieve Python sessie gaan we ervan uit dat het tekstbestand [periodiek_systeem.txt](#) zich in de huidige directory bevindt.

```
>>> symbolen = leesSymbolen('periodiek_systeem.txt')
```

```
>>> langstePrefix('agentia', symbolen)
```

```
'Ag'
```

```
>>> langstePrefix('somatisch', symbolen)
```

```
'S'
```

```
>>> langstePrefix('dichtheid', symbolen)
```

```
''
```

```
>>> chemischWoord('katalyse', symbolen)
```

```
'K-At-Al-Y-Se'
```

```
>>> chemischWoord('basisch', symbolen)
```

```
'Ba-Si-Sc-H'
```

```
>>> chemischWoord('lesie', symbolen)
```

```
''
```