

Rhyming words

Een dichter wil een woordenboek van rijmwoorden aanleggen. Hiervoor beschikt hij over een tekstbestand [c06d](#) waarin woorden aan hun **uitspraak** gekoppeld worden. Het bestand begint met een aantal commentaarregels die starten met een hekje (#). Deze mogen genegeerd worden. Daarna staat op elke regel een woord en zijn uitspraak. De regel

```
PROGRAMMING P R OW1 G R AE2 M IH0 NG
```

geeft bijvoorbeeld aan hoe het woord `programming` moet uitgesproken worden. Het woord wordt hierbij gescheiden van zijn uitspraak door één of meer witruimtekarakters (het woord zelf bevat geen witruimtekarakters), en de uitspraak wordt beschreven door een reeks [fonemen](#) die telkens van elkaar gescheiden worden door één spatie. In totaal wordt er voor het beschrijven van de uitspraak gebruik gemaakt van 39 fonemen (15 klinkers en 24 medeklinkers; fonemen bestaan enkel uit hoofdletters). Elke klinker heeft drie mogelijke **klemtonen** die worden aangegeven door een cijfer op het einde van het foneem (0 indien geen nadruk, 1 indien primaire nadruk en 2 indien secundaire nadruk). In totaal worden er dus 69 symbolen gebruikt om de uitspraak weer te geven.

```
AA odd   AA D
AE at    AE T
AH hut   HH AH T
AO ought AO T
AW cow   K AW
AY hide  HH AY D
B be     B IY
CH cheese CH IY Z
D dee    D IY
DH thee  DH IY
EH Ed    EH D
ER hurt  HH ER T
EY ate   EY T
F fee    F IY
G green  G R IY N
HH he    HH IY
IH it    IH T
IY eat   IY T
JH gee   JH IY
K key    K IY
L lee    L IY
M me     M IY
N knee   N IY
NG ping  P IH NG
OW oat   OW T
OY toy   T OY
P pee    P IY
R read   R IY D
S sea    S IY
SH she   SH IY
T tea    T IY
TH theta TH EY T AH
UH hood  HH UH D
UW two   T UW
V vee    V IY
W we     W IY
```

Y yield Y IY L D
Z zee Z IY
ZH seizure S IY ZH ER

Er zijn echter ook woorden die meerdere **alternatieve uitspraken** hebben. In het tekstbestand wordt elk van deze alternatieven weergegeven op een afzonderlijke regel. De alternatieven worden genummerd door na het woord een getal tussen ronde haakjes te plaatsen. Zo geven de regels

DONKEY D AA1 NG K IY0
DONKEY(2) D AO1 NG K IY0

twee alternatieve uitspraken aan van het woord donkey.

Opgave

De **uitgang** van een uitspraak wordt bepaald als de suffix (substring die loopt tot het einde van de uitspraak) die begint bij de laatste klinker met primaire of secundaire nadruk, of aan het begin van de uitspraak als er geen benadrukte klinkers zijn. Het resterende deel (de prefix voor die laatste klinker) wordt de **stam** van de uitspraak genoemd. De uitspraak P R OW1 G R AE2 M IH0 NG heeft dus stam P R OW1 G R en uitgang AE2 M IH0 NG.

We zeggen dat twee woorden **rijmen** als er een uitspraak bestaat van beide woorden die dezelfde uitgang heeft. Schrijf in eerste instantie een functie `stamuitgang` waaraan een string moet doorgegeven worden die de uitspraak van een woord bevat (fonemen gescheiden door spaties). De functie moet een tuple van twee strings teruggeven, die respectievelijk de stam en de uitgang van de gegeven uitspraak bevatten (fonemen gescheiden door spaties, geen witruimte voor- en achteraan).

Gebruik nu de functie `stamuitgang` om een klasse `Rijm` te schrijven, waarmee kan bepaald worden of twee woorden rijmen of waarmee alle rijmwoorden van een gegeven woord kunnen opgezocht worden. De objecten van de klasse `Rijm` moeten minstens de volgende methoden ondersteunen:

- Een initialisatiemethode waaraan de locatie van een tekstbestand moet doorgegeven worden. Dit bestand bevat de woorden en hun (mogelijks meerdere alternatieve) uitspraken op basis waarvan moet beslist worden of woorden rijmen. De inhoud van het bestand moet opgemaakt zijn volgens de opmaak van het bestand [c06d](#) dat in de inleiding van deze opgave werd besproken.
- Een methode `uitspraken` waaraan een woord moet doorgegeven worden. De methode moet een verzameling met alle uitspraken van het gegeven woord teruggeven.
- Een methode `rijmen` waaraan twee woorden moeten doorgegeven worden. De methode moet een Booleaanse waarde teruggeven die aangeeft of de gegeven woorden rijmen of niet.
- Een methode `rijmwoorden` waaraan een woord moet doorgegeven worden. De methode moet de verzameling met alle woorden teruggeven die rijmen met het gegeven woord (behalve het gegeven woord zelf). Alle woorden in de verzameling moeten in hoofdletters weergegeven worden.

Alle methoden waaraan woorden doorgegeven worden, mogen bij het verwerken van deze woorden geen onderscheid maken tussen hoofdletters en kleine letters. Indien aan een methode een woord wordt doorgegeven waarvan geen uitspraak ingesloten zit in het bestand dat werd

doorgegeven bij het initialiseren van het object van de klasse Rijm, dan moet de methode een AssertionError opwerpen met de boodschap onbekend woord: xxx. Hierbij moet xxx ingevuld worden met het woord waarvoor geen uitspraak kon gevonden worden.

Voorbeeld

```
>>> stamuitgang('S AY1 AH0 N S')
('S', 'AY1 AH0 N S')
>>> stamuitgang('P R OW1 G R AE2 M IH0 NG')
('P R OW1 G R', 'AE2 M IH0 NG')
>>> stamuitgang('AE2 B R AH0 K AH0 D AE1 B R AH0')
('AE2 B R AH0 K AH0 D', 'AE1 B R AH0')
>>> stamuitgang('K AE2 L AH0 F AO1 R N Y AH0')
('K AE2 L AH0 F', 'AO1 R N Y AH0')

>>> dichter = Rijm('c06d')

>>> dichter.uitspraken('science')
{'S AY1 AH0 N S'}
>>> dichter.uitspraken('programming')
{'P R OW1 G R AE2 M IH0 NG'}
>>> dichter.uitspraken('donkey')
{'D AA1 NG K IY0', 'D AO1 NG K IY0'}
>>> dichter.uitspraken('California')
{'K AE2 L AH0 F AO1 R N Y AH0'}
>>> dichter.uitspraken('lambada')
Traceback (most recent call last):
AssertionError: onbekend woord: lambada

>>> dichter.rijmen('science', 'compliance')
True
>>> dichter.rijmen('science', 'programming')
False
>>> dichter.rijmen('science', 'Bryansk')
Traceback (most recent call last):
AssertionError: onbekend woord: Bryansk

>>> dichter.rijmwoorden('science')
{'ALLIANCE', 'COMPLIANCE', 'APPLIANCE', 'RELIANCE', 'PSEUDOSCIENCE', 'DEFIANCE', 'NONCOMPLIANCE'}
>>> dichter.rijmwoorden('abracadabra')
set()
>>> dichter.rijmwoorden('DONKEY')
{'MONKEE', 'HONKY', 'MCCONKEY', 'SWANKY'}
>>> dichter.rijmwoorden('lambada')
Traceback (most recent call last):
AssertionError: onbekend woord: lambada
```