Hy-phen-a-tion

Om te weten waar een woord op het einde van een regel mag gesplitst worden, gebruiken teksteditors een algoritme om te bepalen hoe woorden op een correcte manier in lettergrepen moeten gesplitst worden. Dit algoritme is gebaseerd op een lijst van **patronen** die aangeven waar wel en waar niet mag gesplitst worden. Het patroon 1tion geeft bijvoorbeeld aan dat het woord station moet gesplitst worden als sta-tion, en het patroon t1t geeft aan dat het woord letter moet gesplitst worden als let-ter. In beide gevallen geeft het *oneven* getal 1 aan waar er op een correcte manier mag gesplitst worden.

Er zijn ook uitzonderingen op de splitsingsregels nodig. Het patroon 1tion zou het woord cation immers verkeerdelijk splitsen als ca-tion in plaats van als cat-ion. Daarom bestaan er ook patronen die aangeven op welke plaatsen er *niet* mag gesplitst worden. Zo maakt bijvoorbeeld het patroon .ca2t het voorgaande patroon ongedaan, waardoor de splitsing ca-tion onmogelijk wordt. Hierbij geeft het *even* getal 2 in het patroon aan dat er op deze plaats niet mag gesplitst worden. Een patroon dat begint met een punt geeft aan dat het patroon enkel van toepassing is aan het begin van een woord. Analoog geeft een patroon dat eindigt met een punt aan dat het patroon enkel van toepassing is op het einde van een woord. Uiteraard zijn er ook uitzonderingen op de uitzonderingen mogelijk. Bovendien heeft elke taal zijn eigen splitsingsregels, die vervat zitten in een taalspecifieke lijst van patronen.

Het algoritme voor het splitsen van woorden in lettergrepen dat we in deze opgave zullen gebruiken, werkt tot vijf niveau's diep — splitsen, niet-splitsen, splitsen, niet-splitsen, splitsen. Hierbij worden de eerste en de laatste letter van een woord nooit afzonderlijk afgesplitst. We leggen de werking van het algoritme uit aan de hand van onderstaand voorbeeld, dat aangeeft hoe het woord lettergrepen op een correcte manier in lettergrepen moet gesplitst worden. Op dit woord zijn de patronen r1g, 1te, e1pe, t2er, 4t3t, t5te, .l4, 1gr4 en 4n. van toepassing. Onderstaand schema geeft aan hoe deze patronen gebruikt worden om het woord lettergrepen te splitsen in lettergrepen.

```
. l e t t e r g r e p e n .
          r1g
       1t e
     4t3t
                e1p e
        t2e r
      t5t e
. l4
            1g r4
                   4n .
.0l4e4t5t2e0r1g0r4e1p0e4n0.
  l e t-t e r-g r e-p e n
```

Nadat alle patronen die op het woord van toepassing zijn werden gevonden, bepaalt het *hoogste* cijfer op elke positie tussen twee letters of op die positie mag gesplitst worden of niet (we noemen dit cijfer een **regel**) . Oneven regels geven aan waar er moet gesplitst worden. Met andere woorden, er mogen enkel koppeltekens ingevoegd worden op de positie van de oneven regels. Aanvullend mogen er nooit koppeltekens geplaatst worden voor en na de eerste en laatste letter van het woord. Deze posities worden in bovenstaand schema in het vet aangeduid. Merk ook op dat we in bovenstaand schema een nul geplaatst hebben op alle posities tussen

twee letters waarop geen enkel patroon van toepassing is. Aangezien nullen even zijn, mag hier dus ook niet gesplitst worden.

Opgave

Gevraagd wordt om een klasse Lettergrepen te implementeren waarmee woorden op een correcte manier in lettergrepen kunnen gesplitst worden volgens de specifieke regels van een bepaalde taal. Bij het aanmaken van objecten van deze klasse moet de locatie van een tekstbestand meegegeven worden. Dit bestand moet een lijst van patronen bevatten voor een bepaalde taal, waarbij elk patroon op een afzonderlijke regel staat. Voorts moeten de objecten van deze klasse de volgende methoden ondersteunen:

- Een methode patronen waaraan een woord moet doorgegeven worden dat enkel bestaat uit letters. De methode moet een verzameling teruggeven met alle patronen uit het gegeven bestand die van toepassing zijn op het gegeven woord. Hierbij mag geen onderscheid gemaakt worden tussen hoofdletters en kleine letters, en moeten alle letters die voorkomen in de patronen omgezet worden naar kleine letters.
- Een methode splits waaraan een woord moet doorgegeven worden dat enkel bestaat uit letters. De methode moet het gegeven woord teruggeven, waarbij koppeltekens geplaatst werden op alle posities waar correct mag gesplitst worden. Hierbij moet het gebruik van hoofdletters en kleine letters in het gegeven woord behouden blijven.

Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat het tekstbestand patronen.nl.txt zich in de huidige directory bevindt.

```
>>> nederlands = Lettergrepen('patronen.nl.txt')

>>> nederlands.patronen('lettergrepen')
{'r1g', '1gr4', 'e1pe', 't5te', '1te', '4n.', 't2er', '4t3t', '.l4'}
>>> nederlands.patronen('Babbelaar')
{'.b4', '3b4e', 'a4a4', '3ba', '4bb', '4ar.', '1b', '4r.', 'e1la'}
>>> nederlands.patronen('slaapmutsje')
{'1mu', '2s3je.', '4ts', '4e.', '5slaap', '2p1m', 'a4a4', '5mut', 'sj2', '.s4', 's2l4', 'muts2', 'ts2j', 's1je'}
>>> nederlands.patronen('TROUBADOUR')
{'5dou', 'tr4', '3ba', '1do', '2u2b', '3trou', '1b', '4r.', '.t4'}

>>> nederlands.splits('lettergrepen')
'let-ter-gre-pen'
>>> nederlands.splits('Babbelaar')
'Bab-be-laar'
>>> nederlands.splits('slaapmutsje')
'slaap-muts-je'
>>> nederlands.splits('TROUBADOUR')
'TROU-BA-DOUR'
```

Epiloog

Bronnen

- **Liang FM (1983)**. Word Hy-phen-a-tion by Com-put-er. *Stanford University*.