

SMS language

SMS language is a term for the abbreviations and slang commonly used with mobile phone text messaging. Early mobile phone messaging encouraged users to use abbreviations because *i*) text entry was difficult, requiring multiple key presses on a small keypad to generate each letter, *ii*) messages were limited to a maximal number of characters, and *iii*) operators were charging a cost per letter.

IF YOU POST:	YOU SOUND LIKE:
"Ron Paul is the only candidate who offers us a real choice!"	A TEENAGER
"its gettin 18 so ill b here 4 prob 2 more hrs tops"	A SENATOR

THE INTERNET HAS WOUND UP
IN KIND OF A WEIRD PLACE.

Once SMS language became popular, it took on a life of its own and was often used outside of its original context. At its peak, it was the cause of vigorous debate about its potentially detrimental effect on literacy, but with the advent of alphabetic keyboards on smartphones its use — and the controversies surrounding it — have receded.

Assignment

Some people compress text messages by replacing doubled letters with single letters and by retaining only those vowels that begin a word. Note that the order in which these two operations are executed is important: first removal of duplicates and then removal of vowels. Your task is to implement these operations, such that they can be used to abbreviate text messages. To do so, you have to implement the following four functions that each take a single string argument:

- A function `removeDuplicates` that returns the string resulting from reduction of **successive repetitions** of the same character to the first occurrence of the character. No distinction should be made between upper case and lower case letters in comparing characters.
- A function `removeVowels` that returns the string after removal of all vowels. However, the first character of the string is always retained, even if it is a vowel. As vowels, we consider the letters a, e, i, o and u, both in their upper case and lower case versions.
- A function `txtWord` that returns the string resulting from successively applying the functions `removeDuplicates` and `removeVowels` to the given string.
- A function `txtSentence` that returns the string resulting from applying the function `txtWord` to each word in the given string. In this, words are defined as the longest possible uninterrupted succession of letters. All characters that are not part of a word are simply retained in the sentence.

Example

```
>>> removeDuplicates('bookkeeper')
'bokeper'
>>> removeDuplicates('Aardvark')
'Ardvark'
>>> removeDuplicates('eELGRASS')
'eLGRAS'

>>> removeVowels('bookkeeper')
'bkkpr'
>>> removeVowels('Aardvark')
'Ardvrk'
>>> removeVowels('eELGRASS')
'eLGRSS'

>>> txtWord('Some')
'Sm'
>>> txtWord('people')
'ppl'
>>> txtWord('compress')
'cmprs'
>>> txtWord('text')
'txt'
>>> txtWord('messages')
'msgs'

>>> txtSentence('And now for something completely different!')
'And nw fr smthng cmpltly dfmnt!'
```

```
>>> txtSentence('Some people compress text messages by replacing doubled letters with single letters and by retaining only those vowels that begin a word.')
'Sm ppl cmprs txt msgs by rplcng dbltd ltrs wth snl ltrs and by rtnng only ths vwls tht bgn a wrd.'
```

SMS-taal is een term voor de afkortingen en slang die gebruikt worden bij het versturen van tekstberichten met mobiele telefoons. Toen het versturen van tekstberichten zijn opmars maakte in de wereld van de mobiele telefonie, werd het gebruik van afkortingen in de hand gewerkt doordat *i*) het moeilijk was om tekstberichten in te voeren omdat er verschillende toetsaanslagen nodig waren op een klein toetsenbord om een letter in te voeren, *ii*) tekstberichten vaak beperkt waren tot een maximaal aantal karakters, en *iii*) operatoren een kost per karakter aanrekenden.

IF YOU POST:	YOU SOUND LIKE:
"Ron Paul is the only candidate who offers us a real choice!"	A TEENAGER
"its gettin 18 so ill b here 4 prob 2 more hrs tops"	A SENATOR

THE INTERNET HAS WOUND UP
IN KIND OF A WEIRD PLACE.

Eens SMS-taal ingeburgerd was, begon ze een eigen leven te leiden en werd ze steeds vaker gebruikt buiten haar oorspronkelijke context. Op haar hoogtepunt vormde ze het thema van verhitte discussies over mogelijk schadelijke gevolgen voor de geletterdheid. Door de opkomst van alfanumerieke toetsenborden op smartphones zijn zowel het gebruik van SMS-taal als de controverse die er rond ontstaan is op hun retour.

Opgave

Sommige mensen korten tekstberichten af door dubbele letters te vervangen door enkele letters en door enkel de klinkers over te houden aan het begin van een woord. Merk op dat de volgorde waarin deze twee bewerkingen uitgevoerd worden belangrijk is: eerst dubbels verwijderen en dan pas klinkers verwijderen. We vragen je om deze bewerkingen te implementeren, zodat ze kunnen toegepast worden om tekstberichten af te korten. Hiervoor schrijf je de volgende vier functies, waaraan telkens een string moet doorgegeven worden:

- Een functie `verwijderDubbels` die de string moet teruggeven die bekomen wordt door alle **opeenvolgende herhalingen** van hetzelfde karakter te reduceren tot het eerste voorkomen van dat karakter. Bij het vergelijken van karakters mag geen onderscheid gemaakt worden tussen kleine letters en hoofdletters.
- Een functie `verwijderKlinkers` die de string moet teruggeven die bekomen wordt door alle klinkers te verwijderen. Het eerste karakter van de string blijft echter steeds behouden, ook als dat een klinker zou zijn. Als klinkers beschouwen we de letters a, e, i, o en u, zowel in kleine letters als in hoofdletters.
- Een functie `tkstWoord` die de string moet teruggeven die bekomen wordt door achtereenvolgens de functies `verwijderDubbels` en `verwijderKlinkers` toe te passen op de gegeven string.
- Een functie `tkstZin` die de string moet teruggeven die bekomen wordt door de functie `tkstWoord` toe te passen op elk woord in de gegeven string. Woorden worden daarbij gedefinieerd als de langst mogelijke ononderbroken opeenvolging van letters. Alle karakters die geen deel uitmaken van woorden in de gegeven zin blijven gewoon staan.

Voorbeeld

```
>>> verwijderDubbels('bookkeeper')
'bokeper'
>>> verwijderDubbels('Aardvark')
'Ardvark'
>>> verwijderDubbels('eELGRASS')
'eLGRAS'

>>> verwijderKlinkers('bookkeeper')
'bkkpr'
>>> verwijderKlinkers('Aardvark')
'Ardvrk'
>>> verwijderKlinkers('eELGRASS')
'eLGRSS'

>>> tkstWoord('Some')
'Sm'
>>> tkstWoord('people')
'ppl'
>>> tkstWoord('compress')
'cmprs'
>>> tkstWoord('text')
'txt'
>>> tkstWoord('messages')
'msgs'
```

```
>>> tkstZin('And now for something completely different!')
```

```
'And nw fr smthng cmpltly dfmt!'
```

```
>>> tkstZin('Some people compress text messages by replacing doubled letters with single letters and by retaining only those vowels that begin a word.')
```

```
'Sm ppl cmprs txt msgs by rplcng dbl ltrs wth sngl ltrs and by rtnng only ths vwls tht bgn a wrd.'
```