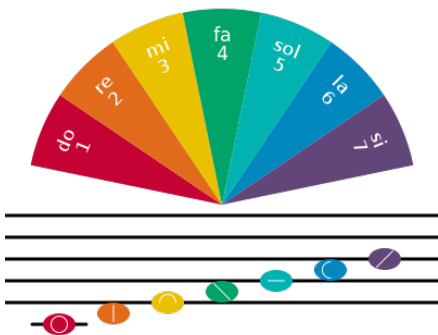


Solresol

[Jean-François Sudre](#) (15 August 1787 – 3 October 1862) had a unique thought in 1817: if people of different cultures can appreciate the same music, why not develop music itself into an international language? The constructed language that he devised based on this idea — which he called **Solresol** — enjoyed a brief spell of popularity, reaching its pinnacle with Boleslas Gajewski's 1902 posthumous publication of *Grammaire du Solrésol*.

Solresol is a musical language based on the principles of solfège, a music education method used to teach the pitch and sight singing. Theoretically, Solresol communication can be done through speaking, singing, flags of different color — even painting. Solresol words are made up of from one to five syllables or notes. Each of these may be one of only seven basic phonemes that correspond to the seven familiar notes of the solfeggio scale: *do, re, mi, fa, sol, la, si*. These phonemes have been used to compose a vocabulary of 2600 roots.



The seven conventional notes, colors, syllables, numerals, and glyphs used to convey solresol phonemes.

Like other constructed languages with *a priori* vocabulary, longer Solresol words are divided into categories of meaning, based on their first syllable. Words beginning with *sol* have meanings related to arts and sciences. Words that begin with *sol**sol* are related to sickness and medicine. Related words thus share initial syllables. For example, *doremi* means day, *dorefa* means week, *doresol* means month and *doredo* the concept of time itself. A unique feature of Solresol is that meanings are negated by reversing the syllables in words. For instance *fala* means good or tasty, and *lafa* means bad.

The teaching of sign languages to the deaf was discouraged between 1880 and 1991 in France, contributing to Solresol's descent into obscurity. After a few years of popularity, Solresol nearly vanished in the face of more successful languages, such as [Volapük](#) and [Esperanto](#). There is still a small community of Solresol enthusiasts scattered across the world, better able to communicate with one another now than previously thanks to the Internet. For example, in the Steven Spielberg movie [Close Encounters of the Third Kind](#) the aliens use Solresol to communicate with the human race.

Preparation

Watch the learning clip about [test-driven development](#) to see how you can make use of doctests. This will allow you to understand the last two statements in the skeleton of the source code that is given in the assignment.

Assignment

You are given the following skeleton of the source code that must lead to the solution of this exercise.

```
def isSolresol(word):
```

```
    """
    Returns a Boolean value that indicates whether or not the given word meets
    the grammar rules of Solresol.
```

```
    >>> isSolresol('doresol')
    True
    >>> isSolresol('doreso')
    False
    >>> isSolresol('salami')
    False
    """
```

```
def shorthand(word):
```

```
    """
    Returns the shorthand notation of the given Solresol word.
```

```
    >>> shorthand('fala')
    'fl'
    >>> shorthand('doremi')
    'drm'
    >>> shorthand('doresol')
    'drso'
    """
```

```

def opposite(word):
    """
    Returns the opposite of the given Solresol word.

    >>> opposite('fala')
    'lafa'
    >>> opposite('doremi')
    'miredo'
    >>> opposite('doresol')
    'solredo'
    """

if __name__ == '__main__':
    import doctest
    print(doctest.testmod())

```

Your task is to implement the three functions in this code fragment, so that the functions meet the following descriptions.

- Write a function `isSolresol` that takes a string argument. The function must return a boolean value that indicates whether or not the given string represents a word that matches the grammar rules of Solresol. Solresol words are made up of from one to five syllables that each may be one of the seven notes of the solfeggio scale: *do, re, mi, fa, sol, la, si* (written using lowercase letters).
- Write a function `shorthand` that takes a Solresol word as its argument. The function must return the shorthand notation of the given word. The syllables of Solresol words are abbreviated in the following way: d[o], r[e], m[i], f[a], so[l], l[a] and s[i].
- Write a function `opposite` that takes a Solresol word as its argument. The function must return the opposite of the given word, which is formed by reversing the syllables in the word.

You may assume that all arguments passed to the functions `shorthand` and `opposite` represent words that match the grammar rules of Solresol, as defined in the description of the function `isSolresol`.

Example

```

>>> isSolresol('doresol')
True
>>> isSolresol('doreso')
False
>>> isSolresol('salami')
False

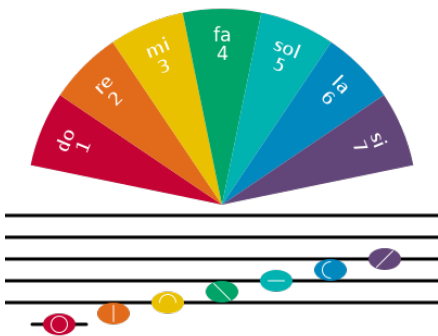
>>> shorthand('fala')
'fl'
>>> shorthand('doremi')
'drm'
>>> shorthand('doresol')
'drso'

>>> opposite('fala')
'lafa'
>>> opposite('doremi')
'miredo'
>>> opposite('doresol')
'solredo'

```

In 1817 kreeg de Fransman [Jean-François Sudre](#) (15 August 1787 – 3 October 1862) het volgende idee: als mensen van verschillende culturen kunnen genieten van dezelfde muziek, waarom zouden we dan geen muziek kunnen gebruiken om er een universele taal mee te ontwikkelen? De kunsttaal die hieruit ontstond — en die hij **Solresol** doopte — genoot enige tijd een grote populariteit, en breikte haar hoogtepunt in 1902 met de publicatie van Boleslas Gajewski's *Grammaire du Solrésol*.

Solresol is een muziktaal die gebaseerd is op het principe van [solfège](#) (de kunst van het ten opzichte van elkaar identificeren van muzikale tonen), en kan zowel worden gesproken en geschreven als gefloten, gezongen, gespeeld of anderszins uitgebeeld. De taal kent slechts zeven lettergrepen die overeenkomen met de zeven noten van de diatonische toonladder: *do*, *re*, *mi*, *fa*, *sol*, *la*, *si*. Met deze noten werd dan een vocabularium samengesteld dat bestond uit 2600 stamwoorden.



De zeven conventionele noten, kleuren, lettergrepen, cijfers en symbolen die gebruikt worden bij de voorstelling van de lettergrepen van Solresol.

Evenals in filosofische talen zijn de solresolwoorden ingedeeld in semantische categorieën, die corresponderen met de eerste lettergreep. Zo hebben woorden die met *sol* beginnen betrekking op kunsten en wetenschappen. Woorden die beginnen met *sol/sol* hebben betrekking op ziekte en medicijnen. *Solresol* zelf betekent taal, en *sol/solredo* betekent migraine. Woorden die met elkaar in verband staan, delen dus dezelfde beginlettergrepen. Zo betekent *doremi* bijvoorbeeld dag, *dorefa* betekent week, *doresol* betekent maand, en *doredo* staat voor het begrip tijd zelf.

Een bijzondere eigenschap van Solresol is dat de betekenis van een woord kan omgekeerd worden door de lettergrepen om te draaien. Zo betekent *fala* goed of lekker, en *lafa* slecht of vies. Andere eigenschappen zijn dat de taal heel eenvoudig te leren en uit te spreken is door analfabeten, en over een eenvoudig maar effectief systeem beschikt dat het mogelijk maakt de functie van een woord binnen een zin te bepalen.

Ondanks aanvankelijk succes moest Solresol het uiteindelijk afleggen tegen succesvollere talen zoals [Esperanto](#). Desondanks is er nog steeds een kleine schare trouwe aanhangers, die voornamelijk via het Internet met elkaar te communiceren. In Steven Spielbergs film [Close Encounters of the Third Kind](#) werd Solresol bijvoorbeeld gebruikt door de aliens om contact te leggen met de mensheid.

Vorbereiding

Bekijk de video over [test-driven development](#) om te leren werken met doctests. Hierdoor zal je de laatste twee statements begrijpen in het skelet van de broncode dat wordt meegegeven met deze opgave.

Opgave

We hebben hieronder reeds het skelet opgemaakt voor de oplossing van deze opgave.

```
def isSolresol(woord):
```

```
    """
    Geeft een Booleaanse waarde terug die aangeeft of het gegeven woord al dan
    niet voldoet aan de grammaticaregels van Solresol.

    >>> isSolresol('doresol')
    True
    >>> isSolresol('doreso')
    False
    >>> isSolresol('salami')
    False
    """
```

```
def afkorting(woord):
```

```
    """
    Geeft de verkorte versie terug van het gegeven solresolwoord.

    >>> afkorting('fala')
    'fl'
    >>> afkorting('doremi')
    'drm'
    >>> afkorting('doresol')
    'drso'
    """
```

```
def antoniem(woord):
```

```
"""
```

Geeft het antoniem terug van het gegeven solresolwoord.

```
>>> antoniem('fala')
'lafa'
>>> antoniem('doremi')
'miredo'
>>> antoniem('doresol')
'solredo'
"""
```

```
if __name__ == '__main__':
    import doctest
    print(doctest.testmod())
```

Je taak bestaat erin om de drie functies uit dit codefragment te implementeren, zodat ze voldoen aan de volgende beschrijvingen.

- Schrijf een functie `isSolresol` waaraan een string moet doorgegeven worden. De functie moet een Booleaanse waarde teruggeven die aangeeft of de gegeven string een woord voorstelt dat voldoet aan de grammaticaregels van Solresol. Solresolwoorden zijn opgebouwd uit één tot vijf lettergrepen die allemaal noten voorstellen (geschreven in kleine letters).
- Schrijf een functie `afkorting` waaraan een solresolwoord moet doorgegeven worden. De functie moet de verkorte versie van het gegeven woord teruggeven. De lettergrepen van solresolwoorden worden op de volgende manier afgekort: `d[o]`, `r[e]`, `m[i]`, `f[a]`, `so[l]`, `l[a]` en `s[i]`.
- Schrijf een functie `antoniem` waaraan een solresolwoord moet doorgegeven worden. De functie moet het antoniem van het gegeven woord teruggeven, dat gevormd wordt door de volgorde van de lettergrepen van het gegeven woord om te keren.

Je mag er van uitgaan dat aan de functies `afkorting` en `antoniem` woorden doorgegeven worden die voldoen aan de grammaticaregels van Solresol, zoals aangegeven bij de omschrijving van de functie `isSolresol`.

Voorbeeld

```
>>> isSolresol('doresol')
True
>>> isSolresol('doreso')
False
>>> isSolresol('salami')
False
```

```
>>> afkorting('fala')
'fl'
>>> afkorting('doremi')
'drm'
>>> afkorting('doresol')
'drso'
```

```
>>> antoniem('fala')
'lafa'
>>> antoniem('doremi')
'miredo'
```

>>> antoniam('doresol')
'solredo'