

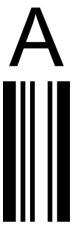
Code 39

Code 39 is a technique for constructing bar codes that was developed in 1974 by Dr. David Allais and Ray Stevens for the company Intermec. With code 39, every character is represented by five vertical bars, that are separated by four vertical white bars. Three of these nine bars are wide and the rest are small. Bar codes of consecutive characters are always separated by a small white bar. The picture below illustrates what the word GEOLOGY looks like as a code 39 bar code.



Code 39 coding of the word GEOLOGY.

Code 39 bar codes are represented in ASCII format by using four symbols that represent the four combinations of small/ wide black/white bars. The four symbols that are used for this are given in the table below.



code 39

bar	symbol
wide-black	B
small-black	S
wide-white	b
small-white	s

Character A, for example, is represented by the sequence BsSsSbSsB.

Assignment

We have assigned the global variable `letter2code` a dictionary, of which every key consists of a character and the corresponding value is a string that represents the translation of the character to its code 39 representation in ASCII format (always a character sequence of length 9). Because we don't distinguish between uppercase and lowercase letters, lowercase letters are not used as a key in this dictionary. Click [here](#) to see the dictionary.

- Write a function `reverse` to which a dictionary should be given as an argument. This function must print a new dictionary as a result that contains the same key/ value pairs as the dictionary that is given as an argument, but where the roles of the keys and values were turned round. You may assume that all values from the dictionary that was given as an argument, are unique.
- Write a function `code39` that converts a given string to its code 39 representation in ASCII format. With this representation, no distinction should be made between uppercase and

lowercase letters, in the sense that both a and A are translated to the string BsSsSbSsB. Two arguments should be given to this function: the string that needs to be converted and the dictionary with characters as keys and the translation of these characters to their code 39 representation in ASCII format as associated values. The function must print a string with the representation in ASCII format as a result. For example, the function must print the string BsBsSbSsSsBsBbSsSsSsBsBsSbSsSsSbBsBsS as a result for the argument c3p0. Note that between the translation of consecutive characters, a letter s must be placed (see the underlined letters in the example), that represents the small white band that is used as a means of separation. If the given string contains characters that can't be represented as a bar code in code 39, the function must print the value None as a result.

- Write a function `decode39` that translates the ASCII format representation of a code 39 bar code to the original text. To this function, two arguments must be given: the string with the ASCII format representation of a code 39 bar code and a dictionary with the ASCII format representation of a code 39 character as a key and the character itself as an associated value. The function must print a string with the original text as a result. For example, the function should print C3PO as a result for the argument BsBsSbSsSsBsBbSsSsSsBsBsSbSsSsSbBsBsS. If the ASCII representation of the bar code contains invalid representations and characters, the function should print the value None as a result.

Example

```
>>> letter2code = {' ': 'SbBsSsBsS', '%': 'SsSbSbSbS', ..., 'Z': 'SbBsBsSsS'}
>>> letter2code['A']
'BsSsSbSsB'
>>> code2letter = reverse(letter2code)
>>> code2letter
{'SsSbBsBsS': '0', 'BsSbSsSsB': '1', 'SbSbSsSbS': '/', ..., 'SsSsSsBbB': 'Q'}
>>> code2letter['BsSsSbSsB']
'A'
>>> code39('c3p0', letter2code)
'BsBsSbSsSsBsBbSsSsSsBsBsSbSsSsSbBsBsS'
>>> decode39('BsBsSbSsSsBsBbSsSsSsBsBsSbSsSsSbBsBsS', code2letter)
'C3P0'
```

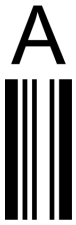
Code 39 is een techniek voor het opstellen van barcodes die in 1974 door Dr. David Allais and Ray Stevens werd ontwikkeld voor het bedrijf Intermecc. Bij code 39 wordt elk karakter voorgesteld door vijf verticale zwarte streken, die van elkaar worden gescheiden door vier verticale witte streken. Drie van deze negen streken zijn breed en de andere zes zijn smal. Barcodes van opeenvolgende karakters worden steeds van elkaar gescheiden door een smalle witte strook. Onderstaande figuur illustreert hoe het woord GEOLOGIE eruit ziet als code 39 barcode.



Code 39 codering van het woord GEOLOGIE.

Code 39 barcodes worden in ASCII-formaat voorgesteld door vier symbolen te gebruiken die de vier combinaties van smalle/brede zwarte/witte banden voorstellen. De vier symbolen die

hiervoor gebruikt worden staan weergegeven in onderstaande tabel.



code 39

strook	symbool
--------	---------

breed-zwart	B
-------------	---

smal-zwart	S
------------	---

breed-wit	b
-----------	---

smal-wit	s
----------	---

Zo wordt het letterteken A bijvoorbeeld voorgesteld door de tekenreeks BsSsSbSsB.

Opgave

Aan de globale variabele `letter2code` hebben we een dictionary toegekend, waarvan elke sleutel bestaat uit een karakter en de bijhorende waarde een string is die de vertaling van het karakter naar zijn code 39 voorstelling in ASCII-formaat (telkens dus een tekenreeks van lengte 9) weergeeft. Omdat we geen onderscheid maken tussen hoofdletters en kleine letters, worden kleine letters niet als sleutel gebruikt in deze dictionary. Klik [hier](#) om de dictionary te bekijken.

- Schrijf een functie `omgekeerd` waaraan een dictionary als argument moet doorgegeven worden. Deze functie moet een nieuwe dictionary als resultaat teruggeven die dezelfde sleutel/waarde paren bevat als de dictionary die als argument wordt doorgegeven, maar waarbij de rollen van sleutels en waarden omgekeerd werden. Je mag er hierbij van uitgaan dat alle waarden van de dictionary die als argument werd doorgegeven uniek zijn.
- Schrijf een functie `code39` die een gegeven string omzet naar zijn code 39 voorstelling in ASCII-formaat. Bij deze omzetting moet geen onderscheid gemaakt worden tussen hoofdletters en kleine letters, in die zin dat zowel de letter a als de letter A worden vertaald naar de string BsSsSbSsB. Aan deze functie moeten twee argumenten doorgegeven worden: de string die moet omgezet worden en een dictionary met karakters als sleutels en de vertaling van deze karakters naar hun code 39 voorstelling in ASCII-formaat als geassocieerde waarden. De functie moet een string met de voorstelling in ASCII-formaat als resultaat teruggeven. Zo moet de functie bijvoorbeeld voor het argument `c3p0` de string BsBsSbSsSsSsBsBbSsSsSsSsSsBsBsSbSsSsSsSbBsBsS als resultaat teruggeven. Let hierbij op het feit dat tussen de vertaling van de opeenvolgende karakters telkens een letter s moet geplaatst worden (zie onderlijnde letters in het voorbeeld), die de smalle witte strook voorstelt die als scheiding gebruikt wordt. Indien de gegeven string karakters bevat die in code 39 niet als barcode kunnen voorgesteld worden, dan moet de functie de waarde `None` als resultaat teruggeven.
- Schrijf een functie `decode39` die de ASCII-formaat voorstelling van een code 39 barcode vertaalt naar de originele tekst. Aan deze functie moeten twee argumenten doorgegeven worden: de string met de ASCII-formaat voorstelling van een code 39 barcode en een dictionary met de ASCII-formaat voorstelling van een code 39 karakter als sleutels en het karakter zelf als geassocieerde waarde. De functie moet een string met de originele tekst

als resultaat teruggeven. Zo moet de functie bijvoorbeeld voor het argument `BsBsSbSsSsBsBbSsSsSsSsBsBsSbSsSsSbBsBsS` de tekenreeks `C3P0` als resultaat teruggeven. Indien de ASCII-voorstelling van de barcode ongeldige voorstellingen van karakters bevat, dan moet de functie de waarde `None` als resultaat teruggeven.

Voorbeeld

```
>>> letter2code = {' ': 'SbBsSsBsS', '%': 'SsSbSbSbS', ..., 'Z': 'SbBsBsSsS'}
>>> letter2code['A']
'BsSsSbSsB'
>>> code2letter = omgekeerd(letter2code)
>>> code2letter
{'SsSbBsBsS': '0', 'BsSbSsSsB': '1', 'SbSbSsSbS': '/', ..., 'SsSsSsBbB': 'Q'}
>>> code2letter['BsSsSbSsB']
'A'
>>> code39('c3p0', letter2code)
'BsBsSbSsSsBsBbSsSsSsSsBsBsSbSsSsSbBsBsS'
>>> decode39('BsBsSbSsSsBsBbSsSsSsSsBsBsSbSsSsSbBsBsS', code2letter)
'C3P0'
```