

# Straddling checkerboard

A **straddling checkerboard** is a type of cipher code that is used to convert an alphabet to numbers. At the same time, there is a certain type of data compression ingrained in this method. For this, a grid (straddling checkerboard) of the following type is used:

	0	1	2	3	4	5	6	7	8	9
	E	T		A	O	N		R	I	S
2	B	C	D	F	G	H	J	K	L	M
6	P	Q	_	U	V	W	X	Y	Z	.

The first row has ten digits in ascending order (0-9). The second line is usually filled with high-frequency letters. Here, two options are left open. Furthermore, this row isn't preceded by a label. The two following rows are labeled with numbers that weren't appointed a letter in the second row, and are filled with the remaining letters of the alphabet.

The symbols of the alphabet can be placed in the grid randomly. The grid has a total of 30 slots, two of which are not being used in the second row. If we use an alphabet of 26 letters, two extra slots can be filled with other symbols. In the example above, we chose a dot (.) and an underscore (\_).

When coding, a letter in the second row is replaced by the number that is used as the column label. Letters in the third and fourth row are replaced by a two-digit number, the first of which is the row label and the second is the column label. Because high frequency letters occur on the second row, the length of the coded message can be kept to a minimum. The message `THE_SHAWSHANK_REDEMPTION`, for example, can be coded using the grid above:

```
T H E _ S H A W S H A N K _ R E D E M P T I O N
1 25 0 62 9 25 3 65 9 25 3 5 27 62 7 0 22 0 29 60 1 8 4 5
```

Decoding is simply done by reversing the process. In spite of the fact that the groups can exist of both 1 and 2 digits, the decoding happens unambiguously because a group of 2 digits always begins with a number that wasn't appointed to a letter in the second row of the grid (2 and 6 in the example above). All other numbers form a group of 1 digit.

## Assignment

Implement the four functions below that are used to code and decode messages using the straddling checkerboard method. To each of these functions, four arguments should be given, the last three of which consist of ten characters and correspond to the second, third and fourth row of the grid used. The second string argument contains two spaces, that indicate the position of the letters in the second row.

- A function `letter2digits` to which as first argument, should be given a character that occurs in the grid given (i.e. in one of the other string arguments). The function has to return the string that consists of one or two digits that correspond with the character given, using the straddling checkerboard method.
- A function `digits2letter` to which, as a first argument, a string consisting of one or two digits should be given. The function must return the character from the given grid that

corresponds to the two digits using the straddling checkerboard method.

- A function `code` to which, as first argument, should be given a string that consists solely of characters from the given grid. The function has to return the coded string (consisting of numbers) that is made using the straddling checkerboard.
- A function `decode` to which, a first argument, should be given a string that consists solely of numbers. The function should return the decoded string that is made using the straddling checkerboard method.

## Example

```
>>> letter2digits('A', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'3'
>>> letter2digits('C', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'21'
>>> letter2digits('U', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'63'

>>> digits2letter('3', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'A'
>>> digits2letter('21', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'C'
>>> digits2letter('63', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'U'

>>> code('THE_SHAWSHANK_REDEMPTION', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'1250629253659253527627022029601845'

>>> decode('1250629253659253527627022029601845', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'THE_SHAWSHANK_REDEMPTION'
```

Een **gebroken dambord** is een vorm van cryptografie die kan gebruikt worden om een alfabet om te zetten naar cijfers. Er zit tegelijkertijd ook een zekere vorm van datacompressie ingebakken in de methode. Hiervoor wordt gebruik gemaakt van een rooster (het gebroken dambord) dat de volgende vorm aanneemt:

	0	1	2	3	4	5	6	7	8	9
	E	T		A	O	N		R	I	S
2	B	C	D	F	G	H	J	K	L	M
6	P	Q	_	U	V	W	X	Y	Z	.

Op de eerste rij staan de tien cijfers in oplopende volgorde (0-9). De tweede rij wordt doorgaans opgevuld met vaak voorkomende letters. Daarbij worden twee posities open gelaten. Deze rij wordt ook niet voorafgegaan door een label. De twee volgende rijen worden telkens gelabeld met de cijfers waaraan geen letter werd toegewezen op de tweede rij, en worden opgevuld met de overige letters van het alfabet.

De symbolen van het alfabet kunnen op een willekeurige manier in het rooster geplaatst worden. Het rooster heeft dus in totaal 30 slots, waarvan er twee niet benut worden op de tweede rij. Als we dus een alfabet gebruiken van 26 hoofdletters, dan kunnen we nog twee extra plaatsen opvullen met andere symbolen. In bovenstaand voorbeeld hebben we als aanvulling gekozen voor een punt (.) en een underscore (\_).

Bij het coderen wordt een letter die op de tweede rij staat vervangen door het cijfer dat als

kolomlabel gebruikt wordt. Letters op de derde en vierde rij worden vervangen door een getal van twee cijfers, waarbij het eerst cijfer het rijlabel is, en het tweede cijfer het kolomlabel. Doordat veel voorkomende letters voorkomen op de tweede rij, kan de lengte van de gecodeerde boodschap kort gehouden worden. De boodschap THE\_SHAWSHANK\_REDEMPTION wordt bijvoorbeeld als volgt gecodeerd op basis van het bovenstaande rooster:

```
T H E _ S H A W S H A N K _ R E D E M P T I O N
1 25 0 62 9 25 3 65 9 25 3 5 27 62 7 0 22 0 29 60 1 8 4 5
```

Decoderen gebeurt eenvoudigweg door het proces om te keren. Ondanks het feit dat de groepen zowel uit één als uit twee cijfers kunnen bestaan, verloopt het decoderen toch ondubbelzinnig omdat een groep van twee cijfers steeds begint met een cijfer dat niet aan een letter gekoppeld is op de tweede rij van het rooster (2 en 6 in bovenstaand voorbeeld). Alle andere cijfers vormen een groep van één cijfer.

## Opgave

Implementeer de volgende vier functies die gebruikt kunnen worden om boodschappen te coderen en decoderen volgens de methode van het gebroken dambord. Aan deze functies moeten telkens vier stringargument doorgegeven worden, waarbij de laatste drie argumenten bestaan uit tien karakters en corresponderen met de twee, derde en vierde rij van het gebruikte rooster. Het tweede stringargument bevat twee spaties, die de posities aangeven waarop geen letters staan op de tweede rij.

- Een functie `letter2cijfers` waaraan als eerste argument een karakter moet doorgegeven worden dat voorkomt in het gegeven rooster (of met ander woorden in één van de drie overige stringargumenten). De functie moet de string bestaande uit één of twee cijfers teruggeven die correspondeert met het gegeven karakter volgens de methode van het gebroken dambord.
- Een functie `cijfers2letter` waaraan als eerste argument een string bestaande uit één of twee cijfers moet doorgegeven worden. De functie moet het karakter uit het gegeven rooster teruggeven dat correspondeert met deze twee cijfers volgens de methode van het gebroken dambord.
- Een functie `codeer` waaraan als eerste argument een string moet doorgegeven worden die uitsluitend bestaat uit karakters die voorkomen in het gegeven rooster. De functie moet de gecodeerde string (bestaande uit cijfers) teruggeven die bekomen wordt door toepassing van de methode van het gebroken dambord.
- Een functie `decodeer` waaraan als eerste argument een string moet doorgegeven worden die uitsluitend bestaat uit cijfers. De functie moet de gedecodeerde string teruggeven die bekomen wordt door toepassing van de methode van het gebroken dambord.

## Voorbeeld

```
>>> letter2cijfers('A', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'3'
>>> letter2cijfers('C', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'21'
>>> letter2cijfers('U', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'63'

>>> cijfers2letter('3', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
'A'
```

```
'A'  
>>> cijfers2letter('21', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
```

```
'C'
```

```
>>> cijfers2letter('63', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
```

```
'U'
```

```
>>> codeer('THE_SHAWSHANK_REDEMPTION', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
```

```
'1250629253659253527627022029601845'
```

```
>>> decodeer('1250629253659253527627022029601845', 'ET AON RIS', 'BCDFGHJKLM', 'PQ_UVWXYZ.')
```

```
'THE_SHAWSHANK_REDEMPTION'
```