

Polybius square

The **Polybius method** is a type of cipher code that was invented by the Greek historian and philosopher Polybius. This method can especially be used to represent a given text as a smaller number of characters. To do so, a rectangle grid is used that is filled with the characters of a given alphabet, written from left to right and from top to bottom in the cells of the grids. This alphabet is called the *long alphabet*. The number of characters in this alphabet must be a complete square. The characters of the second alphabet are written next to the rows (from top to bottom) and above the columns (from left to right) of the square grid. This alphabet is called the *short alphabet*. The square of the number of characters in the short alphabet must be equal to the number of characters in the long alphabet. Scientific characters may occur in both the long and short alphabet.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

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

	S	Q	U	A	R	E
S	A	B	C	D	E	F
Q	G	H	I	J	K	L
U	M	N	O	P	Q	R
A	S	T	U	V	W	X
R	Y	Z	.	,	?	
E	!	;	:	"	&	@

The **Polybius square** can then be used to code a message of which all characters occur in the long alphabet. Every character of the message is converted to two characters of the short alphabet that respectively are next to the row and above the column in which the character of the message occurs in the grid. Decoding a message is done by interpreting both consecutive characters of the decoded message (that consists solely of characters from the short alphabet) as the label of respectively the row and column in which the corresponding character (from the long alphabet) is situated.

Assignment

Implement the four functions below that can be used to code and decode messages according to the Polybius method. To each of these functions, three string should be given. The second and third argument respectively represent the short and long alphabet that are used to build the Polybius square. You may always assume that the length of the long alphabet is equal to the square of the length of the short alphabet. The same character never occurs more than once within an alphabet.

- A function `long2short` to which a character from the long alphabet must be given as the first argument. The function must print a string that consists of the two characters from the short alphabet that are respectively next to the row and above the column in which the character occurs in the Polybius square.
- A function `short2long` to which as a first argument a string should be given that consists of two characters

from the short alphabet. The function must print the character from the long alphabet that is situated in the Polybius square on the row that corresponds with the first character of the given string and the column that corresponds with the second character.

- A function `code` to which as a first argument a string should be given that consists solely of characters from the long alphabet. The function must print the encrypted string based on the Polybius method.
- A function `decode` to which as a first argument a string should be given that consists of an even number of characters from the short alphabet. The function must print a string based on the Polybius method.

Example

```
>>> long2short('P', '12345', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
'35'
>>> long2short('P', '123456', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')
'34'
>>> long2short('P', 'SQUARE', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!:;&@')
'UA'

>>> short2long('35', '12345', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
'P'
>>> short2long('34', '123456', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')
'P'
>>> short2long('UA', 'SQUARE', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!:;&@')
'P'

>>> code('POLYBIUS', '12345', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
'3534315412244543'
>>> code('POLYBIUS', '123456', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')
'3433265112234341'
>>> code('POLYBIUS VIERKANT', 'SQUARE', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!:;&@')
'UAUUQERSSQUAUASRUAAQUSRUEQRSSUQAQ'

>>> decode('3534315412244543', '12345', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
'POLYBIUS'
>>> decode('3433265112234341', '123456', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')
'POLYBIUS'
>>> decode('UAUUQERSSQUAUASRUAAQUSRUEQRSSUQAQ', 'SQUARE', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!:;&@')
'POLYBIUS VIERKANT'
```

De **methode van Polybius** is een vorm van cryptografie die werd uitgevonden door de Griekse historicus en wijsheer Polybius. In het bijzonder kan de methode gebruikt worden om een gegeven tekst voor te stellen met een kleiner aantal karakters. Hierbij wordt gebruik gemaakt van een vierkant rooster dat wordt opgevuld met de karakters van een gegeven alfabet, die van links naar rechts en van boven naar onder in de cellen van het roosters worden uitgeschreven. We noemen dit alfabet het *lange alfabet* en het aantal karakters van dit alfabet moet dus een volkomen kwadraat zijn. De karakters van een tweede alfabet worden dan naast de rijen (van boven naar onder) en boven de kolommen (van links naar rechts) van het vierkant rooster geschreven. We noemen dit alfabet het *korte alfabet*. Het kwadraat van het aantal letters in het korte alfabet moet gelijk zijn aan het aantal letters in het lange alfabet. Er mogen gemeenschappelijke karakters voorkomen in het korte en het lange alfabet.

	1	2	3	4	5
1	A	B	C	D	E
2	F	G	H	I	K
3	L	M	N	O	P
4	Q	R	S	T	U
5	V	W	X	Y	Z

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

	S	Q	U	A	R	E
S	A	B	C	D	E	F
Q	G	H	I	J	K	L
U	M	N	O	P	Q	R
A	S	T	U	V	W	X
R	Y	Z	.	,	?	
E	!	;	:	"	&	@

Het **Polybiusvierkant** kan dan gebruikt worden voor het coderen van een boodschap waarvan alle karakters voorkomen in het lange alfabet. Elk karakter van de boodschap wordt daarbij omgezet naar de twee karakters van het korte alfabet die respectievelijk naast de rij en boven de kolom staan waarop het karakter van de boodschap voorkomt in het rooster. Het decoderen van een boodschap gebeurt door elke twee opeenvolgende karakters van de gedecodeerde boodschap (die uitsluitend bestaat uit karakters van het korte alfabet) te interpreteren als het label van respectievelijk de rij en de kolom waarop het corresponderende karakter (uit het lange alfabet) staat.

Opgave

Implementeer de volgende vier functies die gebruikt kunnen worden om boodschappen te coderen en decoderen volgens de methode van Polybius. Aan deze functies moeten telkens drie strings doorgegeven worden. Het tweede en derde argument stellen respectievelijk het korte en het lange alfabet voor die gebruikt worden om het Polybiusvierkant op te bouwen. Je mag er steeds van uitgaan dat de lengte van het lange alfabet gelijk is aan het kwadraat van de lengte van het korte alfabet. Hetzelfde karakter komt ook nooit meerdere keren voor binnen een alfabet.

- Een functie `lang2kort` waaraan als eerste argument een karakter moet doorgegeven worden dat voorkomt in het lange alfabet. De functie moet een string teruggeven die bestaat uit de twee karakters van het korte alfabet die in het Polybiusvierkant respectievelijk naast de rij en boven de kolom staan waarop het gegeven karakter voorkomt.
- Een functie `kort2lang` waaraan als eerste argument een string moet doorgegeven worden die bestaat uit twee karakters van het korte alfabet. De functie moet het karakter van het lange alfabet teruggeven dat zich in het Polybiusrooster bevindt op de rij die correspondeert met het eerste karakter van de gegeven string en de kolom die correspondeert met het tweede karakter van de gegeven string.
- Een functie `codeer` waaraan als eerste argument een string moet doorgegeven worden die uitsluitend bestaat uit karakters van het lange alfabet. De functie moet de gecodeerde string teruggeven die bekomen wordt door toepassing van de methode van Polybius.
- Een functie `decodeer` waaraan als eerste argument een string moet doorgegeven worden die bestaat uit een even aantal karakters van het korte alfabet. De functie moet de gedecodeerde string teruggeven die bekomen wordt door toepassing van de methode van Polybius.

Voorbeeld

```
>>> lang2kort('P', '12345', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
'35'
>>> lang2kort('P', '123456', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')
'34'
>>> lang2kort('P', 'SQUARE', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!;"&@')
'UA'
```

```
>>> kort2lang('35', '12345', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
'P'
>>> kort2lang('34', '123456', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')
'P'
>>> kort2lang('UA', 'SQUARE', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!;:"&@')
'P'

>>> codeer('POLYBIUS', '12345', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
'3534315412244543'
>>> codeer('POLYBIUS', '123456', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')
'3433265112234341'
>>> codeer('POLYBIUS VIERKANT', 'SQUARE', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!;:"&@')
'UAUUQERSSQQUAUASRUAAQUSRUEQRSSUQAQ'

>>> decodeer('3534315412244543', '12345', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ')
'POLYBIUS'
>>> decodeer('3433265112234341', '123456', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789')
'POLYBIUS'
>>> decodeer('UAUUQERSSQQUAUASRUAAQUSRUEQRSSUQAQ', 'SQUARE', 'ABCDEFGHIJKLMNOPQRSTUVWXYZ .,?!;:"&@')
'POLYBIUS VIERKANT'
```