

Pigpen cipher

James Leeson (1756–1794) is buried in New York's Trinity Church Cemetery. His tombstone bears a curious inscription. It remained a local puzzle until 1889, when the *Trinity Record* discovered that it reads as REMEMBER DEATH encoded in pigpen cipher.



Tombstone of James Leeson (1756-1794) in New York's Trinity Church Cemetery. Note that the cipher on top of the stone makes use of a slightly different key compared to the one used in this exercise: the wrinkle is that J is omitted, because in colonial orthography it was identical with I. All letters following J in the alphabet therefore shift one position forward in the grids of the key used for encoding and decoding.

The **pigpen cipher** (sometimes referred to as the **masonic cipher** or **Freemason's cipher**) is a geometric simple substitution cipher that exchanges letters for symbols which are fragments of a grid. The exact origin of the cipher is uncertain, but records of this system have been found which go back to at least the 18th century. Variations of this cipher were used by both the Rosicrucian brotherhood and the Freemasons, though the latter used it so often that the system is frequently called the Freemason's cipher. They began using it in the early 18th century to keep their records of history and rites private, and for correspondence between lodge leaders.

The oldest known version is from Heinrich Cornelius Agrippa von Nettesheim who explains the method in his *De occulta philosophia* published in 1533. This work was reused in 1586 by Blaise de Vigenère in his treatise about cryptography. There are many possible variations, but any trained crypto-analyst will usually have no trouble deciphering the encoded messages.

Assignment

Your task is to decode a given word according to the pigpen cipher that makes use of the following key.

A	B	C
D	E	F
G	H	I

J	K	L
M	N	O
P	Q	R

S	T	U
V	W	X
Y	Z	

This is done by substituting each letter of the word by its symbolic representation in the pigpen cipher. This representation forms a 3 × 3 grid of characters. The middle character of the symbolic representation indicates what grid of the key contains the letter: a space represents the left grid, a dot (.) the middle grid, and a colon (:) the grid to the right. In addition, lines must be drawn to the left, right, top and bottom according to the position of the letter in the grid. As an example, the symbolic representation of all letters in the middle grid are depicted below.

```

|  ||  |  --+  ++  +-  --+  ++  +-
.|  |.|  |.  .|  |.|  |.  .|  |.|  |.
--+  ++  +-  --+  ++  +-  |  ||  |
J   K   L   M   N   O   P   Q   R

```

Note that fragments of a horizontal line are thus represented by a dash (-) and fragments of a vertical line by a pipe symbol (|). Corner points of the symbolic representation where two lines meet are represented by a plus sign (+). Your task:

- Write a function `pigpenletter` that takes a single letter as an argument. The function must return a string containing three lines of three characters. These characters form the symbolic representation of the given letter in pigpen notation. Make sure that the number of conditions that need to be tested is kept to the absolute minimum when determining the representation of the given letter in pigpen notation. You definitely have to avoid programming constructs of the form: if the letter is A then its symbolic representation is the following, else if the letter is B then its symbolic representation is the following, ...
- Use the function `pigpenletter` to write a function `pigpen`. The function takes a string that only contains letters and must return a string that consists of three lines that contain the symbolic representation of the given word in pigpen notation. This is done by separating the symbolic representations of the individual letters with a single column containing three spaces.

Both functions must handle the given string in a case insensitive way.

Example

```

>>> print(pigpenletter('A'))
|
|
--+
>>> print(pigpenletter('m'))
--+
.|
--+
>>> print(pigpenletter('U'))
|

```

```
|:  
+--
```

```
>>> print(pigpen('python'))  
--+ --+ || ++ +-- ++  
. | :|:|:|:| |.  
| | ++ || +-- ++  
>>> print(pigpen('Rosenkreutz'))  
+-- +-- |++ ++ ||+-- ++ | || ++  
|. |. :|:|:|:|:| |:|:|:|:|  
| +-- --+ ++ ++ ++ | ++ +-- ++ ||
```

Op het Trinity Church Cemetery in New York ligt James Leeson (1756–1794) begraven. Bovenaan zijn grafsteen staat een merkwaardige inscriptie, waarvan pas in 1889 bleek dat ze de tekst REMEMBER DEATH voorstelt in het rozenkruisersgeheimschrift.



Grafsteen van James Leeson (1756-1794) op het Trinity Church Cemetery in New York. Merk op dat het geheimschrift opgesteld is op basis van een variant van de sleutel die in deze opgave gebruikt wordt: de letters I en J werden op dezelfde manier geschreven in de koloniale spelling en dus werd de letter J weggelaten uit de sleutel. Alle letters die na de J komen, schuiven daardoor één positie naar voor in de roosters die gebruikt worden voor de versleuteling.

H e t **rozenkruisersgeheimschrift** (ook wel **vrijmetselaarsalfabet** of **maçonniek kwadratschrift**) is een monoalfabetische versleutelingsmethode waarbij de letters van de platte tekst gecodeerd worden door middel van eenvoudig te onthouden diagrammen. De methode — die ook onder de minder lovende naam **varkenshokversleuteling** (of **pigpen**) bekend staat — wordt in het algemeen toegeschreven aan de rozenkruisers. Dit geheim genootschap van wijsgerigen zou er in de 18e eeuw gebruik van gemaakt hebben om er haar geheimen mee te bewaren.

De oudst bekende variant is van Heinrich Cornelius Agrippa von Nettesheim die de methode uitlegt in zijn *De occulta philosophia* uit 1533. Dit werk werd in 1586 door Blaise de Vigenère hergebruikt in zijn tractaat aangaande geheimschrift. Er zijn vele varianten mogelijk, maar een geoefend cryptanalist zal doorgaans geen enkele moeite hebben met de ontcijfering.

Opgave

De opgave bestaat erin een gegeven woord uit te schrijven, maar dan gecodeerd in het rozenkruisersgeheimschrift dat gebruik maakt van de volgende sleutel.

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

Hierbij moet elke letter van het gegeven woord vertaald worden naar zijn symbolische voorstelling in het rozenkruisersgeheimschrift. Deze voorstelling bestaat uit 3×3 karakters. Het middelste karakter van de symbolische voorstelling geeft aan in welk van de drie roosters de letter voorkomt: een spatie stelt het linkse rooster voor, een punt (.) het middelste rooster, en een dubbelpunt (:) het rechtse rooster. Voorts moeten er lijnen links, rechts, boven en onder getrokken worden, overeenkomstig de positie van de letter in het rooster. Hieronder tonen we bij wijze van voorbeeld de symbolische voorstelling van de letters in het middelste rooster.

```

|  ||  |  --+  ++  +-  --+  ++  +-
.|  |.|  |  .|  |.|  |  .|  |.|  |
--+  ++  +-  --+  ++  +-  |  ||  |
J   K   L   M   N   O   P   Q   R

```

Merk dus op dat fragmenten van een horizontale lijn voorgesteld worden door een koppelteken (-) en fragmenten van een verticale lijn door een verticale streep (|). Hoekpunten van de symbolische voorstelling waar twee lijnen samenkomen, worden voorgesteld door een plusteken (+). Gevraagd wordt:

- Schrijf een functie `rozenkruisersletter` waaraan een letter moet doorgegeven worden. De functie moet een string teruggeven bestaande uit 3 regels van 3 karakters, die de gegeven letter in het rozenkruisersgeheimschrift voorstelt. Zorg ervoor dat er zo weinig mogelijk voorwaarden moeten getest worden om de voorstelling van de letter in het rozenkruisersgeheimschrift te bepalen. Vermijd met andere woorden programmeerconstructies van de vorm: als de letter A is dan is dit de symbolische voorstelling, als de letter B is dan is dit de voorstelling, ...
- Gebruik de functie `rozenkruisersletter` om een functie `rozenkruisersgeheimschrift` te schrijven. Aan deze functie moet een woord doorgegeven worden dat enkel bestaat uit letters. De functie moet een string bestaande uit drie regels teruggeven, die de symbolische voorstelling van het gegeven woord in het rozenkruisersgeheimschrift voorstelt. Hierbij wordt tussen de symbolische voorstelling van de letters telkens een kolom van drie spaties gevoegd.

Beide functies mogen geen onderscheid maken tussen hoofdletters en kleine letters.

Voorbeeld

```

>>> print(rozenkruisersletter('A'))
|
|
--+
>>> print(rozenkruisersletter('m'))
--+
.|

```

```
--+
>>> print(rozenkruisersletter('U'))
|
|:
+--
```

```
>>> print(rozenkruisersgeheimschrift('python'))
```

```
--+ --+ || ++ +-- ++
```

```
.| :|:|:|:|. |.
```

```
| | ++ || +-- ++
```

```
>>> print(rozenkruisersgeheimschrift('Rosenkreutz'))
```

```
+-- +-- | ++ ++ || +-- ++ | || ++
```

```
|. |. :|:|:|:|:|. ||: |:|:|
```

```
| +-- --+ ++ ++ ++ | ++ +-- ++ ||
```