

# In memoriam

Epitaph of John Laird McCaffery (1940-1995), who is buried in the Mount Royal Cemetery in Montreal (Canada):



The text was written jointly by his ex-wife and his mistress. Read the first letter of each line ...

## Preparation

Python has a built-in function `isinstance(object, data type)`. The function returns a Boolean value, that indicates whether the given object belongs to the given data type.

```
>>> value = 1
>>> type(value)
<class 'int'>
>>> isinstance(value, int)
True
>>> isinstance(value, float)
False
>>> isinstance(value, list)
False
```

## Assignment

An **acrostic** is a poem or other form of writing in which the first letter, syllable or word of each line, paragraph or other recurring feature in the text spells out a word or a message. Asked:

- Write a function `selection` to which two arguments must be passed: a string that contains a line of a poem and a sequence of numbers  $n_1, n_2, \dots, n_m$  as a list or tuple, where  $n_i \in \mathbb{N}_0$  ( $i = 1, 2, \dots, m$ ) applies. The function must return the string that is formed by the characters  $k_1 k_2 \dots k_m$ . Where  $k_i$  ( $i = 1, 2, \dots, m$ ) represents the character at position  $n_i$  in the given line. When determining the position of a character in a line, the white space is ignored (spaces, tabs and new lines) and the first character that is not a white character is in position 1. If the line contains less than  $n_i$

(niet-witruimte) characters, then  $k_i$  is ignored.

- Use the function `selection` to write a function `acrostic` to which a file name must be passed. The file name refers to the text file that contains the lines of a poem. The function also has a second optional parameter `positions` to which a sequence of integers  $n_1, n_2, \dots, n_m$  can be passed as a list or tuple, where  $n_i \in \mathbb{N}_{>0}$  ( $i = 1, 2, \dots, m$ ) applies. If this sequence counts only one integer  $n_1$ , then the value  $n_1$  can also be passed as integer to the parameter `positions`. The default value of the parameter `positions` is 1. The function should return the string that is formed by selecting the characters in the given positions of each line of the poem (cf. function `selection`) and putting all these characters after each other.

## Example

In the following example session we assume that the files [in\\_memoriam.txt](#) and [maude.txt](#) are in the current directory.

```
>>> selection('Free your body', [1])
'F'
>>> selection('Unfold your powerful wings', [5, 12])
'lo'
>>> selection('Climb up the highest mountains', [15, 6, 11])
'euh'
>>> selection('Kick your feet up in the air', [500, 250, 10, 2])
'ej'

>>> acrostic('in_memoriam.txt')
'FUCKYOU'
>>> acrostic('in_memoriam.txt', 1)
'FUCKYOU'
>>> acrostic('in_memoriam.txt', positions=23)
'say'
>>> acrostic('in_memoriam.txt', positions=[4, 7, 12])
'euyoyompikutmnverheye'
>>> acrostic('maude.txt', positions=[1, 2])
'PeCuLiArAcRoStlc'
```

Grafschrift van John Laird McCaffery (1940-1995), die in Montreal (Canada) begraven ligt in het Mount Royal Cemetery:



De tekst werd gezamenlijk opgesteld door zijn ex-vrouw en zijn minares. Lees de eerste letter van elke regel...

## Opgave

Een gedicht waarvan bepaalde — meestal de eerste — letters van iedere regel of strofe achter elkaar gelezen zelf ook een woord of zin vormen, wordt een **acrostichon** (ook: *naamdicht* of *lettervers*) genoemd. Gevraagd wordt:

- Schrijf een functie `selectie` waaraan twee argumenten moeten doorgegeven worden: een string die een versregel van een gedicht bevat en een reeks getallen  $n_1, n_2, \dots, n_m$  als een lijst of een tuple, waarbij geldt dat  $n_i \in \mathbb{N}_0$  ( $i = 1, 2, \dots, m$ ). De functie moet de string teruggeven die gevormd wordt door de karakters  $k_1 k_2 \dots k_m$ . Hierbij stelt  $k_i$  ( $i = 1, 2, \dots, m$ ) het karakter voor op positie  $n_i$  in de gegeven versregel. Bij het bepalen van de positie van een karakter in de versregel wordt witruimte (spaties, tabs en regeleindes) genegeerd, en staat het eerste karakter dat geen witruimte is op positie 1. Indien de gegeven versregel minder dan  $n_i$  (niet-witruimte) karakters bevat, dan wordt  $k_i$  genegeerd.
- Gebruik de functie `selectie` om een functie `acrostichon` te schrijven waaraan een bestandsnaam moet doorgegeven worden. Deze bestandsnaam verwijst naar een tekstbestand dat de versregels van een gedicht bevat. De functie heeft ook nog een tweede optionele parameter `posities` waaraan een reeks getallen  $n_1, n_2, \dots, n_m$  kan doorgegeven worden als een lijst of een tuple, waarbij geldt dat  $n_i \in \mathbb{N}_0$  ( $i = 1, 2, \dots, m$ ). Indien deze reeks uit slechts één getal  $n_1$  bestaat, dan mag de waarde  $n_1$  ook als integer doorgegeven worden aan de parameter `posities`. De standaardwaarde van de parameter `posities` is 1. De functie moet de string teruggeven die gevormd wordt door van elke versregel van het gedicht de karakters op de aangegeven posities te selecteren (cf. functie `selectie`) en al deze karakters achter elkaar te zetten.

## Voorbereiding

Om na te gaan of een gegeven object een bepaald gegevenstype heeft, kan je natuurlijk gebruik maken van de ingebouwde functie `type(o)` die het gegevenstype van het object `o` teruggeeft. Het is

echter beter om hiervoor de ingebouwde functie `isinstance(o, t)` te gebruiken. Deze functie geeft een Booleaanse waarde terug die aangeeft of het object `o` al dan niet behoort tot het type `t`.

```
>>> type(3) == int
True
>>> isinstance(3.14, int)
False
>>> isinstance(3.14, float)
True
>>> isinstance([1, 2, 3], list)
True
```

## Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat de tekstbestanden [in\\_memoriam.txt](#) en [maude.txt](#) zich in de huidige directory bevinden.

```
>>> selectie('Free your body', [1])
'F'
>>> selectie('Unfold your powerful wings', [5, 12])
'lo'
>>> selectie('Climb up the highest mountains', [15, 6, 11])
'euh'
>>> selectie('Kick your feet up in the air', [500, 250, 10, 2])
'ej'

>>> acrostichon('in_memoriam.txt')
'FUCKYOU'
>>> acrostichon('in_memoriam.txt', 1)
'FUCKYOU'
>>> acrostichon('in_memoriam.txt', posities=23)
'say'
>>> acrostichon('in_memoriam.txt', posities=[4, 7, 12])
'euyoyompikutmnverheye'
>>> acrostichon('maude.txt', posities=[1, 2])
'PeCuLiArAcRoStlc'
```