

Word sums

If we value each letter corresponding to its position in the alphabet, we may compute the value of a word as the sum of the values of its individual letters. We can use this to hunt for word sums. A **word sum** is a sequence of (usually two) words whose sum of word values equals the word value of another word that can be associated with the sequence of words.

KING + CHAIR = THRONE

Checking whether two words are a word sum when combined with a third word is fairly straightforward. However, it is far more difficult to find good word sums, since especially the associations determine their quality. Here we provide you with a couple of examples, where we give the word value of each word between a pair of brackets.

term 1	term 2	result
ARM (32)	BEND (25)	ELBOW (57)
WHITE (65)	HOUSE (68)	GOVERNMENT (133)
MONA (43)	LISA (41)	LEONARDO (84)
PETER (64)	PAN (31)	NEVERLAND (95)
FAMILY (66)	TREE (48)	ANCESTORS (114)
RED (27)	BULL (47)	COCKTAIL (74)
EGG (19)	PLANT (63)	AUBERGINE (82)
ANT (35)	LION (50)	DOODLEBUG (85)
VISUAL (84)	BASIC (34)	MICROSOFT (118)
BLACK (29)	JACK (25)	VEGAS (54)

Along the same lines we can also make associations with names of famous people.

term 1	term 2	result
JOHN (47)	CLEESE (49)	HUMOUR (96)
TOM (48)	HANKS (53)	FORREST (101)
BOB (19)	MARLEY (74)	RASTAFARI (93)
KURT (70)	COBAIN (44)	NOVOSELIC (114)
NELSON (79)	MANDELA (50)	HUMANITARIAN (129)
EMMA (32)	WATSON (92)	VOLDEMORT (124)
JAMES (48)	BOND (35)	DANIEL CRAIG (83)
GEORGE (57)	LUCAS (56)	JAR JAR BINKS (113)
STEPHEN (87)	HAWKING (73)	TEXT TO SPEECH (160)
CLOCKWORK (111)	ORANGE (60)	STANLEY KUBRICK (171)

With the last four examples in the above table, we only take into account the letters to determine the value of a word. This allowed us to come up with multi-word results.

Assignment

- Write a function `lettervalue` that takes a string containing a single character. The function must return the value of the character. Letters have a value that corresponds to their position in the alphabet. No distinction is made between uppercase and lowercase letters. All other characters have a zero value.
- Write a function `wordvalue` that takes a string as its argument. The function must return the sum of the values of all characters in the given string.
- Write a function `wordsum` that takes three strings. The function must return a Boolean value that indicates whether or not the sum of the word values of the first two arguments equals the word value of the third argument.

Example

```
>>> lettervalue('A')
1
>>> lettervalue('j')
10
>>> lettervalue('!')
0

>>> wordvalue('arm')
32
>>> wordvalue('BEND')
25
>>> wordvalue('elbow')
57

>>> wordsum('arm', 'BEND', 'elbow')
True
>>> wordsum('KING', 'chair', 'THRONE')
True
>>> wordsum('Monty', 'Python', 'SHRUBBERRY')
False
```

Als we elke letter een waarde geven overeenkomstig met zijn positie in het alfabet, dan kunnen we de waarde van een woord berekenen als de som van de waarden van de letters in het woord. Daarmee kunnen we op zoek gaan naar woordsommen. Een **woordsom** is een reeks (van meestal twee) woorden waarvan de som van de woordwaarden gelijk is aan de woordwaarde van een woord dat ermee kan geassocieerd worden.

KING + CHAIR = THRONE

Controleren of twee woorden een woordsom vormen met een derde woord is heel eenvoudig. Het is echter veel moeilijker om goede woordsommen te vinden. Het zijn immers voornamelijk de associaties die de kwaliteit van de woordsommen bepalen. Hier zijn er alvast nog een paar, waarbij we na elk woord tussen ronde haakjes ook telkens zijn woordwaarde vermelden.

term 1	term 2	resultaat
ARM (32)	BEND (25)	ELBOW (57)
WHITE (65)	HOUSE (68)	GOVERNMENT (133)
MONA (43)	LISA (41)	LEONARDO (84)
PETER (64)	PAN (31)	NEVERLAND (95)
FAMILY (66)	TREE (48)	ANCESTORS (114)

term 1	term 2	resultaat
BEND (27)	BULL (47)	COCKTAIL (74)
EGG (19)	PLANT (63)	AUBERGINE (82)
ANT (35)	LION (50)	DOODLEBUG (85)
VISUAL (84)	BASIC (34)	MICROSOFT (118)
BLACK (29)	JACK (25)	VEGAS (54)

Op dezelfde manier kunnen we ook associaties maken met namen van bekende personen.

term 1	term 2	resultaat
JOHN (47)	CLEESE (49)	HUMOUR (96)
TOM (48)	HANKS (53)	FORREST (101)
BOB (19)	MARLEY (74)	RASTAFARI (93)
KURT (70)	COBAIN (44)	NOVOSELIC (114)
NELSON (79)	MANDELA (50)	HUMANITARIAN (129)
EMMA (32)	WATSON (92)	VOLDEMORT (124)
JAMES (48)	BOND (35)	DANIEL CRAIG (83)
GEORGE (57)	LUCAS (56)	JAR JAR BINKS (113)
STEPHEN (87)	HAWKING (73)	TEXT TO SPEECH (160)
CLOCKWORK (111)	ORANGE (60)	STANLEY KUBRICK (171)

Bij de laatste voorbeelden in bovenstaande tabel houden we enkel rekening met de letters om de woordwaarde te bepalen. Op die manier konden we ook resultaten bekomen die uit meerdere woorden bestaan.

Opgave

- Schrijf een functie `letterwaarde` waaraan een string moet doorgegeven worden die bestaat uit één enkel karakter. De functie moet de waarde van het karakter teruggeven. Letters hebben een waarde overeenkomstig hun positie in het alfabet. Hierbij wordt geen onderscheid gemaakt tussen hoofdletters en kleine letters. Alle andere karakters hebben waarde nul.
- Schrijf een functie `woordwaarde` waaraan een string moet doorgegeven worden. De functie moet de som van de waarden van alle karakters in de string teruggeven.
- Schrijf een functie `woordsom` waaraan drie strings moeten doorgegeven worden. De functie moet een Booleaanse waarde teruggeven die aangeeft of de som van de woordwaarden van de eerste twee strings al dan niet gelijk is aan de woordwaarde van de derde string.

Voorbeeld

```
>>> letterwaarde('A')
1
>>> letterwaarde('j')
10
>>> letterwaarde('!')
0
```

```
>>> woordwaarde('arm')
32
>>> woordwaarde('BEND')
```

25

```
>>> woordwaarde('elbow')
```

57

```
>>> wordsom('arm', 'BEND', 'elbow')
```

True

```
>>> wordsom('KING', 'chair', 'THRONE')
```

True

```
>>> wordsom('Monty', 'Python', 'SHRUBBERY')
```

False