

Autogram

Autograms (Greek: αὐτός = self, γράμμα = letter) are sentences that describes themselves in the sense of providing an inventory of its their characters. As a result, they are also called self-enumerating or self-documenting sentences. They have been invented by British engineer Lee Sallows, who also coined the word autogram. An essential feature is the use of full cardinal number names such as "one", "two", ... in recording character counts. Often, letter counts only are recorded while punctuation signs are ignored, as in this example:

This sentence employs two a's, two c's, two d's, twenty-eight e's, five f's, three g's, eight h's, eleven i's, three l's, two m's, thirteen n's, nine o's, two p's, five r's, twenty-five s's, twenty-three t's, six v's, ten w's, two x's, five y's, and one z.

The first autogram to be published was composed by Sallows in 1982 and appeared in Douglas Hofstadter's *Metamagical Themas* column in *Scientific American*.

Only the fool would take trouble to verify that his sentence was composed of ten a's, three b's, four c's, four d's, forty-six e's, sixteen f's, four g's, thirteen h's, fifteen i's, two k's, nine l's, four m's, twenty-five n's, twenty-four o's, five p's, sixteen r's, forty-one s's, thirty-seven t's, ten u's, eight v's, eight w's, four x's, eleven y's, twenty-seven commas, twenty-three apostrophes, seven hyphens and, last but not least, a single !

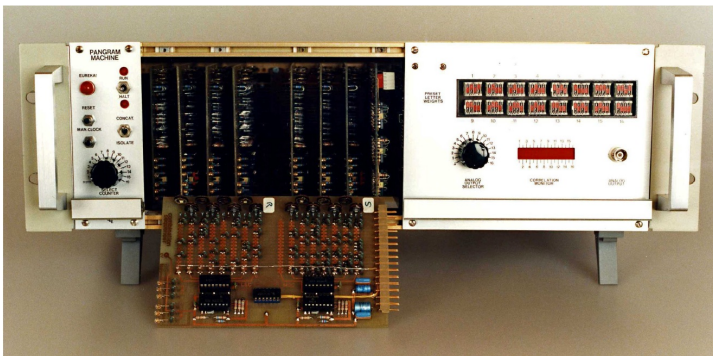
The task of producing an autogram is perplexing because the object to be described cannot be known until its description is first complete. This is why it's so extraordinary that poet Rudy Kousbroek has been able to come up with a perfect translation in Dutch. Sure enough a heavenly translation of Lee Sallows' masterpiece.

Alleen 'n ezel gelooft dat ik de moeite heb genomen na te tellen dat deze zin bestaat uit zestien a's, drie b's, vier c's, vijftien d's, negenenzeventig e's, zes f's, veertien g's, zes h's, vierendertig i's, vier j's, zeven k's, acht l's, vijf m's, achtendertig n's, zes o's, vier p's, een q, achttien r's, vijfendertig s's, drieëndertig t's, drie u's, dertien v's, drie w's, een x, een y, tien z's, plus dertig komma's, vierentwintig afkappingstekens, twee aanhalingstekens en, niet te vergeten, een enkel !

A type of autogram that has attracted special interest is the **autogramic pangram**, a self-enumerating sentence in which every letter of the alphabet occurs at least once. Certain letters do not appear in the three autograms above, which are therefore not pangrams. The first ever self-enumerating pangram appeared in a Dutch newspaper and was composed by Rudy Kousbroek.

Dit pangram bevat vijf a's, twee b's, twee c's, drie d's, zeseneventig e's, vijf f's, vier g's, twee h's, vijftien i's, vier j's, een k, twee l's, twee m's, zeventien n's, een o, twee p's, een q, zeven r's, vierentwintig s's, zestien t's, een u, elf v's, acht w's, een x, een y, en zes z's.

Sallows, who lives in the Netherlands, was challenged by Kousbroek to produce a self-enumerating translation of this pangram into English — an impossible-seeming task. This prompted Sallows to construct an electronic Pangram Machine.



One of the eight printed circuit cards of the Pangram Machine, bearing two counters and the resistor fan-outs for letters R and S. At left, two groups of three LEDs signal current counter positions.

Eventually the machine succeeded, producing the example below which was published in *Scientific American* in October 1984.

This pangram contains four a's, one b, two c's, one d, thirty e's, six f's, five g's, seven h's, eleven i's, one j, one k, two l's, two m's, eighteen n's, fifteen o's, two p's, one q, five r's, twenty-seven s's, eighteen t's, two u's, seven v's, eight w's, two x's, three y's, & one z.

Miguel Lerma of the *Universidad Politecnica* of Madrid discovered/invented the following self-inventorying Spanish pangram:

Esta frase contiene exactamente doscientas treinta y cinco letras: veinte a's, una b, dieciseis c's, trece d's, treinta e's, dos f's, una g, una h, diecinueve i's, una j, una k, dos l's, dos m's, veintidos n's, catorce o's, una p, una q, diez r's, treinta y tres s's, diecinueve t's, doce u's, cinco v's, una w, dos x's, cuatro y's, y dos z's.

Assignment

- Write a function `letterFrequencies` that takes a string argument. The function must return a dictionary that maps each letter that occurs in the given sentence onto the number of occurrences of that letter in the sentence. In counting letter occurrences, the function may not make a distinction between uppercase and lowercase letters, and all keys of the returned dictionary must be lowercase letters.

All characters in the given sentence that are not letters must be ignored during the construction of the dictionary.

- Write a function `letterPositions` that takes a string argument. The function must return a dictionary that maps each letter that occurs in the given sentence onto the set of all positions where the letter occurs in the sentence. In doing so, the function may not make a distinction between uppercase and lowercase letters, and all keys of the returned dictionary must be lowercase letters. All characters of the sentence must be taken into account when determining the positions of the letters in the sentence. The first character is at position 0, the second character at position 1, and so on.

Example

```
>>> frequentie = letterFrequencies("fifteen e's, seven f's, four g's, six h's, eight i's, four n's, five o's, six r's, eighteen s's, eight t's, four u's, three v's, two w's, three x's")
>>> frequentie['e']
15
>>> frequentie['f']
7
>>> frequentie['g']
4
```

```
>>> frequentie = letterFrequencies("sixteen e's, five f's, three g's, six h's, nine i's, five n's, four o's, six r's, eighteen s's, eight t's, three u's, three v's, two w's, four x's")
>>> frequentie['e']
16
>>> frequentie['f']
5
>>> frequentie['g']
3
```

```
>>> positions = letterPositions("fifteen e's, seven f's, four g's, six h's, eight i's, four n's, five o's, six r's, eighteen s's, eight t's, four u's, three v's, two w's, three x's")
>>> positions['e']
{97, 67, 4, 5, 8, 43, 141, 14, 142, 16, 83, 88, 89, 121, 122}
>>> positions['f']
{0, 64, 2, 108, 19, 54, 24}
>>> positions['g']
{99, 29, 45, 85}
```

```
>>> positions = letterPositions("sixteen e's, five f's, three g's, six h's, nine i's, five n's, four o's, six r's, eighteen s's, eight t's, three u's, three v's, two w's, four x's")
>>> positions['e']
{96, 4, 5, 8, 46, 110, 16, 111, 82, 87, 56, 88, 26, 27, 121, 122}
>>> positions['f']
{18, 138, 13, 53, 63}
>>> positions['g']
{98, 84, 29}
```

Epilogue

A few years ago Lee Sallows began to wonder whether it's possible to produce a sentence that reckons its totals as percentages. This is more difficult, because the percentages won't always work out to be integers. As he worked on the problem he mentioned it to a few others, among them British computer scientist Chris Patuzzo. And a few days ago, Patuzzo sent him this:

This sentence is dedicated to Lee Sallows and to within one decimal place four point five percent of the letters in this sentence are a's, zero point one percent are b's, four point three percent are c's, zero point nine percent are d's, twenty point one percent are e's, one point five percent are f's, zero point four percent are g's, one point five percent are h's, six point eight percent are i's, zero point one percent are j's, zero point one percent are k's, one point one percent are l's, zero point three percent are m's, twelve point one percent are n's, eight point one percent are o's, seven point three percent are p's, zero point one percent are q's, nine point nine percent are r's, five point six percent are s's, nine point nine percent are t's, zero point seven percent are u's, one point four percent are v's, zero point seven percent are w's, zero point five percent are x's, zero point three percent are y's and one point six percent are z's.

More details can be found [here](#). The next challenge is a version where the percentages are accurate to two decimal places — Patuzzo is working on that now.

Resources

- **Sallows L (1985)**. In Quest of a Pangram. *Abacus* **2(3)**, 22-40. [↗](#)
- **Hofstadter DR (1996)**. Metamagical Themas: Questing for the Essence of Mind and Pattern. *Basic Books*, 390-392. ISBN 978-0-465-04566-2. [↗](#)
- **Letaw JR (1985)**. Pangrams: A Nondeterministic Approach. *Abacus* **2(3)**, 42-47.
- **Kousbroek R (1983)**. Welke vraag heeft vierendertig letters? *NRC Handelsblad*, Cultureel Supplement 640, 11 februari 1983, p.3.
- **Kousbroek R (1983)**. Instructies voor het demonteren van een bom. *NRC Handelsblad*, Cultureel Supplement 644, 11 maart 1983, p.9.
- **Kousbroek R (1984)**. De Logologische Ruimte. *Meulenhof*, 135-153. [↗](#)
- **Dewdney AK (1984)**. Computer Recreations. *Scientific American*, 18-22. [↗](#)

Autogrammen (Grieks: αὐτός = zelf, γράμμα = letter) zijn zinnen die zichzelf beschrijven, in die zin dat ze een inventaris maken van hun eigen letters. Daarom worden ze ook wel eens zelfbeschrijvende of zelfdocumenterende zinnen genoemd. Ze werden uitgevonden door de Britse ingenieur Lee Sallows, die er ook de term autogram voor bedacht. Essentieel bij het opstellen van autogrammen is het

gebruik van voluit geschreven hoofdtelwoorden zoals "een", "twee", ... bij het neerschrijven van de letterfrequenties. Bij het tellen worden vaak alleen maar de letters in rekening gebracht terwijl de leestekens genegeerd worden, zoals in dit voorbeeld:

This sentence employs two a's, two c's, two d's, twenty-eight e's, five f's, three g's, eight h's, eleven i's, three l's, two m's, thirteen n's, nine o's, two p's, five r's, twenty-five s's, twenty-three t's, six v's, ten w's, two x's, five y's, and one z.

Het eerste gepubliceerde autogram werd samengesteld door Sallows in 1982 en verscheen in het tijdschrift *Scientific American* in de column *Metamagical Themas* van Douglas Hofstadter.

Only the fool would take trouble to verify that his sentence was composed of ten a's, three b's, four c's, four d's, forty-six e's, sixteen f's, four g's, thirteen h's, fifteen i's, two k's, nine l's, four m's, twenty-five n's, twenty-four o's, five p's, sixteen r's, forty-one s's, thirty-seven t's, ten u's, eight v's, eight w's, four x's, eleven y's, twenty-seven commas, twenty-three apostrophes, seven hyphens and, last but not least, a single !

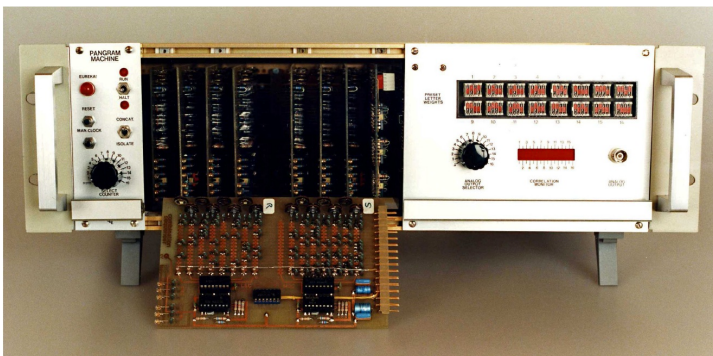
De uitdaging om een autogram te maken is ontzettend moeilijk omdat de zin die moet beschreven worden onbekend is totdat de omschrijving zelf voltooid is. Daarom is het des te verwonderlijker dat de Nederlander Rudy Kousbroek erin slaagde om er een perfecte vertaling van te maken. Zowaar een hemelse vertaling van Lee Sallow's meesterwerk.

Alleen 'n ezel gelooft dat ik de moeite heb genomen na te tellen dat deze zin bestaat uit zestien a's, drie b's, vier c's, vijftien d's, negenenzeventig e's, zes f's, veertien g's, zes h's, vierendertig i's, vier j's, zeven k's, acht l's, vijf m's, achtendertig n's, zes o's, vier p's, een q, achttien r's, vijfendertig s's, drieëndertig t's, drie u's, dertien v's, drie w's, een x, een y, tien z's, plus dertig komma's, vierentwintig afkappingstekens, twee aanhalingstekens en, niet te vergeten, een enkel !

Een speciaal soort autogram die bijzondere aandacht heeft gekregen is het **autogrammatisch pangram**: een zelfbeschrijvende zin waarin elke letter van het alfabet minstens één keer voorkomt. Sommige letters komen niet voor in de bovenstaande drie autogrammen, waardoor het geen pangrammen zijn. Het eerste autogrammatisch pangram was van de hand van Rudy Kousbroek en verscheen in het NRC Handelsblad.

Dit pangram bevat vijf a's, twee b's, twee c's, drie d's, zesenzeventig e's, vijf f's, vier g's, twee h's, vijftien i's, vier j's, een k, twee l's, twee m's, zeventien n's, een o, twee p's, een q, zeven r's, vierentwintig s's, zestien t's, een u, elf v's, acht w's, een x, een y, en zes z's.

Sallows, die in Nederland woont, werd door Kousbroek uitgedaagd om een zelfbeschrijvende Engelse vertaling van dit pangram te maken — een schier onmogelijke taak. Sallows deed beroep op zijn ingenieursachtergrond om speciaal voor deze uitdaging een computer te ontwikkelen, die hij de naam Pangram Machine gaf.



Een van de acht printkaarten die door Pangram Machine gebruikt worden. De printkaart bevat twee tellers en de weerstandsuitgangen voor de letters R en S. Links zie je twee groepen van drie LEDs die de huidige positie van de tellers aangeven.

Uiteindelijk slaagde de Pangram Machine in zijn opzet, en produceerde onderstaand voorbeeld dat in oktober 1984 gepubliceerd werd in *Scientific American*.

This pangram contains four a's, one b, two c's, one d, thirty e's, six f's, five g's, seven h's, eleven i's, one j, one k, two l's, two m's, eighteen n's, fifteen o's, two p's, one q, five r's, twenty-seven s's, eighteen t's, two u's, seven v's, eight w's, two x's, three y's, & one z.

Miguel Lerma van de *Universidad Politecnica* van Madrid slaagde er ook in om een autogrammatisch pangram te construeren in het Spaans:

Esta frase contiene exactamente doscientas treinta y cinco letras: veinte a's, una b, dieciseis c's, trece d's, treinta e's, dos f's, una g, una h, diecinueve i's, una j, una k, dos l's, dos m's, veintidos n's, catorce o's, una p, una q, diez r's, treinta y tres s's, diecinueve t's, doce u's, cinco v's, una w, dos x's, cuatro y's, y dos z's.

Opgave

- Schrijf een functie letterfrequenties waaraan een string moet doorgegeven worden. De functie moet een dictionary teruggeven die elke letter die voorkomt in de gegeven zin afbeeldt op het aantal voorkomens van die letter in de zin. Hierbij mag de functie bij het tellen van de letters geen onderscheid maken tussen hoofdletters en kleine letters, en moeten alle sleutels van de dictionary kleine letters zijn. Alle karakters uit de gegeven zin die geen letter zijn, moeten genegeerd worden bij het opstellen van de

dictionary.

- Schrijf een functie `letterfrequenties` waaraan een string moet doorgegeven worden. De functie moet een dictionary teruggeven die elke letter die voorkomt in de gegeven zin afbeeldt op de verzameling van alle posities waar die letter voorkomt in de zin. Hierbij mag de functie geen onderscheid maken tussen hoofdletters en kleine letters, en moeten alle sleutels van de dictionary kleine letters zijn. Bij het bepalen van de posities van de letters moeten alle karakters van de zin in rekening gebracht worden. Hierbij staat het eerste karakter van de zin op positie 0, het tweede karakter op positie 1, enzoverder.

Voorbeeld

```
>>> frequentie = letterfrequenties("fifteen e's, seven f's, four g's, six h's, eight i's, four n's, five o's, six r's, eighteen s's, eight t's, four u's, three v's, two w's, three x's")
>>> frequentie['e']
15
>>> frequentie['f']
7
>>> frequentie['g']
4
```

```
>>> frequentie = letterfrequenties("sixteen e's, five f's, three g's, six h's, nine i's, five n's, four o's, six r's, eighteen s's, eight t's, three u's, three v's, two w's, four x's")
>>> frequentie['e']
16
>>> frequentie['f']
5
>>> frequentie['g']
3
```

```
>>> posities = letterposities("fifteen e's, seven f's, four g's, six h's, eight i's, four n's, five o's, six r's, eighteen s's, eight t's, four u's, three v's, two w's, three x's")
>>> posities['e']
{97, 67, 4, 5, 8, 43, 141, 14, 142, 16, 83, 88, 89, 121, 122}
>>> posities['f']
{0, 64, 2, 108, 19, 54, 24}
>>> posities['g']
{99, 29, 45, 85}
```

```
>>> posities = letterposities("sixteen e's, five f's, three g's, six h's, nine i's, five n's, four o's, six r's, eighteen s's, eight t's, three u's, three v's, two w's, four x's")
>>> posities['e']
{96, 4, 5, 8, 46, 110, 16, 111, 82, 87, 56, 88, 26, 27, 121, 122}
>>> posities['f']
{18, 138, 13, 53, 63}
>>> posities['g']
{98, 84, 29}
```

Epiloog

Een paar jaar geleden begon Lee Sallows zich af te vragen of het ook mogelijk zou zijn om een zin in elkaar te knutselen die het aantal voorkomens van de letters omschrijft in percentages. Dit is nog moeilijker omdat de percentages niet altijd natuurlijke getallen zijn. Terwijl hij dit idee aan het uitwerken was, sprak hij er ook met enkele anderen over. Onder andere met de Britse computerwetenschapper Chris Patuzzo. Een paar dagen geleden stuurde Patuzzo hem het volgende door:

This sentence is dedicated to Lee Sallows and to within one decimal place four point five percent of the letters in this sentence are a's, zero point one percent are b's, four point three percent are c's, zero point nine percent are d's, twenty point one percent are e's, one point five percent are f's, zero point four percent are g's, one point five percent are h's, six point eight percent are i's, zero point one percent are j's, zero point one percent are k's, one point one percent are l's, zero point three percent are m's, twelve point one percent are n's, eight point one percent are o's, seven point three percent are p's, zero point one percent are q's, nine point nine percent are r's, five point six percent are s's, nine point nine percent are t's, zero point seven percent are u's, one point four percent are v's, zero point seven percent are w's, zero point five percent are x's, zero point three percent are y's and one point six percent are z's.

Meer details vind je [hier](#). De volgende uitdaging is een versie waarvan de percentages nauwkeurig zijn tot op twee cijfers na de komma — Patuzzo is daar momenteel aan aan het werken.

Bronnen

- **Sallows L (1985)**. In Quest of a Pangram. *Abacus* **2(3)**, 22-40. [↗](#)
- **Hofstadter DR (1996)**. Metamagical Themas: Questing for the Essence of Mind and Pattern. *Basic Books*, 390-392. ISBN 978-0-465-04566-2. [↗](#)
- **Letaw JR (1985)**. Pangrams: A Nondeterministic Approach. *Abacus* **2(3)**, 42-47.
- **Kousbroek R (1983)**. Welke vraag heeft vierendertig letters? *NRC Handelsblad*, Cultureel Supplement 640, 11 februari 1983, p.3.
- **Kousbroek R (1983)**. Instructies voor het demonteren van een bom. *NRC Handelsblad*, Cultureel Supplement 644, 11 maart 1983, p.9.
- **Kousbroek R (1984)**. De Logologische Ruimte. *Meulenhof*, 135-153. [↗](#)
- **Dewdney AK (1984)**. Computer Recreations. *Scientific American*, 18-22. [↗](#)