

# ESP game

Today it is still very hard to determine what is in a picture using a computer program. Image recognition is a problem computers are almost unable to solve. To the human brain, this problem is fairly simple, but then again it is very difficult to put a large number of people to work in order to process picture that the Google search engine finds on the internet.

To solve this issue, the [ESP game](#) was developed. The idea behind this computer game is to use the intelligence of the human brain to perform a task computers can't do without effort. It was originally developed by Luis von Ahn of Carnegie Mellon University. Google bought a license 2006 to improve the search results for online pictures.



After logging in, you are linked to a random fellow player. The players don't know the identity of the other fellow players and they can't communicate with one another. Both players see a picture. The goal is to type the same word as your fellow player. In order to get to that word, both players have to sum up a number of words that make them think of the picture. This way, the system learns what objects can be seen in the picture. As soon as both players have typed the same word, they get to see a new picture. They have half a minute to describe 15 photos in this way.

For some of the words, there is a list of taboo words, that the players are not allowed to use. These are the words of which the system already knows they can be linked to the pictures. Making the game more difficult by excluding certain words, teaches words to the system that can be linked to the photo in second or third order.

## Assignment

The goal of this assignment is to implement a part of the ESP game, with which a list of taboo words can be determined that will be shown to the players, based on a list of words that the

system has already linked to the picture after the game.

- Write a function `tabooLength` that can be used to determine the amount of words that will be shown to both players. To this function a list of words must be given as a first obligatory argument. These are the words that the system has already linked to a picture. Also, it is possible to give values to this function for the optional parameters `minimum` and/or `maximum`. The function must print a random amount of integers that is situated between the minimum and maximum (boundaries included). If no minimum is given, or if the minimum is smaller than zero, the minimum length equals zero. If no maximum is given, or if the maximum is larger than the number of words in the list given, the maximum length equals the amount of words from the list. If the value `None` is given for the minimum or maximum, the default value should be used.
- Use the function `tabooLength` to write a function `tabooWords` that prints a list with a random selection of `$k$` different words from a given word list. The words selected must be printed as an alphabetically ordered list. A list of words must be given to the function as a first, obligatory argument. These are words that were already linked to a picture. Also, it is possible to give values to the function for the optional parameters `minimum` and/or `maximum`. The number of words `$k$` of the result list should always be in between the minimum and maximum (boundaries included). If no minimum is given, or if the minimum is smaller than zero, the minimum length equals zero. If no maximum is given, or if the maximum is larger than the number of words in the list given, the maximum length equals the amount of words from the list. If the value `None` is given for the minimum or maximum, the default value should be used.

**Tip:** Try using functions from the `random` module when implementing the functions above. This way you don't have to reinvent the wheel.

## Example

```
>>> words = ['forest', 'meadow', 'scenery', 'hills']
>>> tabooLength(words)
3
>>> tabooLength(words, minimum = 2)
4
>>> tabooLength(words, maximum = 3)
1
>>> tabooLength(words, minimum = 2, maximum = 3)
3
>>> tabooLength(words, minimum = -2, maximum = 6)
1

>>> tabooWords(words)
['forest', 'hills', 'meadow', 'scenery']
>>> tabooWords(words, minimum = 2)
['forest', 'meadow', 'scenery']
>>> tabooWords(words, maximum = 3)
['forest', 'hills', 'meadow']
>>> tabooWords(words, minimum = 2, maximum = 3)
['hills', 'meadow', 'scenery']
>>> tabooWords(words, minimum = -2, maximum = 6)
['forest', 'hills', 'meadow', 'scenery']
```

Vandaag de dag is het nog steeds zeer moeilijk om aan de hand van computerprogramma's te

achterhalen wat er precies op een foto te zien is. Beeldherkenning is immers een probleem waar computers bijna niet toe in staat zijn. Voor het menselijk brein is dit een vrij eenvoudig probleem, maar het is dan weer zeer moeilijk om veel mensen aan het werk te zetten om bijvoorbeeld alle foto's te verwerken die de Google zoekrobot op het Internet aantreft.

Om hieraan tegemoet te komen werd de [ESP game](#) ontwikkeld. De idee achter dit computerspelletje is om de intelligentie van het menselijk brein spelenderwijs in te zetten om een taak uit te voeren waartoe computers niet in staat zijn. Het werd oorspronkelijk ontwikkeld door Luis von Ahn van Carnegie Mellon University. Google kocht in 2006 een licentie van het spelletje om de zoekresultaten voor online afbeeldingen te kunnen verbeteren.



Nadat je bent ingelogd, wordt je automatisch gekoppeld aan een willekeurige medespeler. De medespelers kennen elkaars identiteit niet en kunnen ook niet met elkaar communiceren. Beide medespelers krijgen dezelfde foto te zien. Het doel is om zo snel mogelijk een gemeenschappelijk woord in te typen, en om dit te bekomen zullen beide spelers natuurlijk woorden opsommen die ze linken aan de afbeelding. Op die manier leert het systeem dat achter dit spelletje zit welke objecten er op de foto te zien zijn. Van zodra beide spelers eenzelfde woord hebben opgegeven, krijgen ze een nieuwe afbeelding te zien. Ze hebben twee en een halve minuut om op die manier 15 foto's te omschrijven.

Bij sommige foto's wordt ook een lijst van taboewoorden opgegeven, die de spelers niet mogen gebruiken. Dit zijn de woorden waarvan het systeem uit eerdere spelletjes reeds weet dat ze veelvuldig met de foto gelinkt worden. Door het spelletje moeilijker te maken door enkele gangbare woorden uit te sluiten, leert het systeem ook woorden die slechts in twee of derde orde aan de foto gelinkt worden.

## Opgave

Het doel van deze opgave bestaat erin om een stukje van het ESP game te implementeren,

waarmee een lijst van taboewoorden kan bepaald worden die aan de spelers zal getoond worden, op basis van een lijst van woorden die het systeem achter het spelletje reeds aan een foto gelinkt heeft.

- Schrijf een functie `taboeLengte` die kan gebruikt worden om te bepalen hoeveel taboewoorden beide spelers zullen te zien krijgen. Aan deze functie moet als eerste verplicht argument een lijst van woorden doorgegeven worden. Dit zijn de woorden die het systeem reeds gelinkt heeft aan een foto. Aan de functie kunnen ook nog waarden doorgegeven worden voor de optionele parameters `minimum` en/of `maximum`. De functie moet als resultaat een willekeurig aantal teruggeven dat tussen het opgegeven minimum en maximum gelegen is (grenzen inbegrepen). Indien geen minimum wordt opgegeven, of indien het opgegeven minimum kleiner is dan nul, dan is de minimumlengte gelijk aan nul. Indien geen maximum wordt opgegeven, of indien het maximum groter is dan het aantal woorden in de opgegeven lijst, dan is de maximumlengte gelijk aan het aantal woorden in de opgegeven lijst. Indien voor het minimum of het maximum de waarde `None` wordt opgegeven, dan moet de standaardwaarde gebruikt worden.
- Gebruik de functie `taboeLengte` om een functie `taboeWoorden` te schrijven die een lijst met een willekeurige selectie van `$k$` verschillende woorden uit een gegeven woordenlijst als resultaat teruggeeft. De geselecteerde woorden moeten als een alfabetische gesorteerde lijst teruggegeven worden. Aan de functie moet als eerste verplicht argument een lijst van woorden doorgegeven worden. Dit zijn de woorden die het systeem reeds gelinkt heeft aan een foto. Aan de functie kunnen ook nog waarden doorgegeven worden voor de optionele parameters `minimum` en/of `maximum`. Het aantal woorden `$k$` van de resultaatlijst moet steeds tussen het opgegeven minimum en maximum gelegen zijn (grenzen inbegrepen). Indien geen minimum wordt opgegeven, of indien het opgegeven minimum kleiner is dan nul, dan is de minimumlengte gelijk aan nul. Indien geen maximum wordt opgegeven, of indien het maximum groter is dan het aantal woorden in de opgegeven lijst, dan is de maximumlengte gelijk aan het aantal woorden in de opgegeven lijst. Indien voor het minimum of het maximum de waarde `None` wordt opgegeven, dan moet de standaardwaarde gebruikt worden.

**Tip:** Maak zoveel mogelijk gebruik van functies uit de `random` module bij de implementatie van de bovenstaande functies. Op die manier hoef je het warm water niet opnieuw uit te vinden.

## Voorbeeld

```
>>> woorden = ['forest', 'meadow', 'scenery', 'hills']
>>> taboeLengte(woorden)
3
>>> taboeLengte(woorden, minimum = 2)
4
>>> taboeLengte(woorden, maximum = 3)
1
>>> taboeLengte(woorden, minimum = 2, maximum = 3)
3
>>> taboeLengte(woorden, minimum = -2, maximum = 6)
1

>>> taboeWoorden(woorden)
['forest', 'hills', 'meadow', 'scenery']
>>> taboeWoorden(woorden, minimum = 2)
['forest', 'meadow', 'scenery']
```

```
>>> taboeWoorden(waarden, maximum = 3)
['forest', 'hills', 'meadow']
>>> taboeWoorden(waarden, minimum = 2, maximum = 3)
['hills', 'meadow', 'scenery']
>>> taboeWoorden(waarden, minimum = -2, maximum = 6)
['forest', 'hills', 'meadow', 'scenery']
```