Dans ce TP, vous allez utiliser la bibliothèque Gensim couramment utilisée dans le traitement automatique des langues et en particulier dans le plongement lexical des mots.

I. Utiliser un modèle word2vec existant

a. Vous allez d'abord utiliser un modèle pré-entrainé qui existe sur genism
Utiliser la bonne fonction pour le télécharger : word2vec-google-news-300

Note : le téléchargement prendra souvent quelques minutes.

b. Word2vec est entraîné sur Wikipédia. Vérifier si votre prénom est représenté par un vecteur.
c. Quel est le mot le plus similaire à votre prénom ? Cela vous paraît-il logique ?
d. Le mot "computer" est-il similaire au mot "software" ?
e. Quels sont les 5 mots les plus similaires au mot "computer" ?
f. Parmi ces mots ["computer", "software", "intelligence", "engineering"], trouver l'intrus (celui qui ne correspond pas vraiment) ?
g. A l'aide d'une opération sémantique, trouver :
   - La monnaie indienne
   - Le participe passé du verbe « buy ».
   - La capitale du Chili
   - Le nom de famille du président chinois
   - La femelle du cheval

II. Entraîner votre propre modèle depuis la page Wikipédia « computer science ».

a. Appliquer ce code sur la page wikipedia https://en.wikipedia.org/wiki/Computer

```
import bs4 as bs
import urllib.request
import re
import nltk
scrapped_data = urllib.request.urlopen('https://fr.wikipedia.org/wiki/Informatique')
wiki_info = scrapped_data.read()
print(wiki_info)
```

b. Parser ce texte xml avec : *bs.BeautifulSoup(wiki_info,'lxml')*
c. Utiliser le bon paramètre dans la fonction *find_all('bon_parametre')* afin d'extraire les paragraphes de cet article.
d. Extraire le texte brut et comparer le avec le texte dans : https://en.wikipedia.org/wiki/Computer
e. Segmenter le texte en phrases puis en mots
f. Vous allez maintenant enlever les mots non porteurs de sens (stop_words) du texte. Pour cela :
   - Vérifier si le package "*stopwords*" existe dans votre machine sinon télécharger le à l'aide de la commande : *nltk.download('stopwords')*
   - Afficher les mots anglais non porteurs de sens.

```
from nltk.corpus import stopwords
print(stopwords.words('english'))
```

- Enlever maintenant les « stop words » de la liste des mots segmentés.

g. A partir de ces mots entrainer un modèle word2vec.
h. Quels sont les 5 mots les plus similaires au mot « computer ».
i. Quelle est la similarité du mot “computer” et le mot “software” ?
Comparer le résultat avec celui obtenu par le modèle pré-entraîné (partie I)

III. Analyse de sentiments

a. Utiliser la bibliothèque NLTK pour afficher la polarité du mot “confortable”

```
from nltk.sentiment import SentimentIntensityAnalyzer
sia = SentimentIntensityAnalyzer()
sia.polarity_scores('comfortable')
```

b. En utilisant le modèle, écrire un programme qui permet de représenter les phrases suivantes par des vecteurs. Vous pouvez utiliser la moyenne des vecteurs des mots qui les composent multipliés par les polarités des mots.
   - *The atmosphere and staff were very friendly*
   - *I wanted to order room service however they won’t offering the service in the morning*
   - *The temperature of the room was hard to regulate*
   - *Great hotel but unfortunate receptionist*
   - *It was alright but could have been better*
   - *I liked the layout of the room*

c. Proposer une fonction qui fait la classification des commentaires en positif et négatif.