TP Python 6 - Analyse des mots de la langue française

Le fichier `mots.txt` contient la plupart des mots en français. Les mots sont séparés par des points-virgules. Ils sont constitués exclusivement de lettres minuscules et non accentuées. Il n'y a pas de mots composés.

1) Créer une liste de tous les mots du fichier (utiliser `split`). Combien y en a-t-il ?

2) Quel est le mot le plus long et combien a-t-il de lettres ?

3) Combien y a-t-il de mots de 1 lettre, de 2 lettres, etc. ?

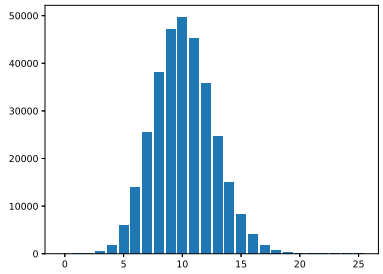
```
Il y a 2 mots de 1 lettres
Il y a 81 mots de 2 lettres
Il y a 427 mots de 3 lettres
...
```

On peut aussi donner le résultat sous forme de dictionnaire :

```
{1: 2, 2: 81, 3: 427, 4: 1799, 5: 5891, 6: 13902, 7: 25456, 8: 38096, 9: 47249,
 10: 49688, 11: 45225, 12: 35713, 13: 24772, 14: 15007, 15: 8212, 16: 4066,
 17: 1874, 18: 817, 19: 353, 20: 157, 21: 68, 22: 27, 23: 7, 24: 2, 25: 1}
```

On peut enfin représenter le résultat par un histogramme à l'aide de la fonction `bar` du module `matplotlib.pyplot` de la manière suivante (où `nb = [2, 81, 427, ...]` est la liste des nombres de mots des différentes longueurs) :

```
import matplotlib.pyplot as plt
plt.bar(list(range(1, 26)), nb)
plt.show()
```



4) Combien y a-t-il de mots commençant par `a`, par `b`, etc. ?

```
Il y a 23161 mots qui commencent par a
Il y a 14341 mots qui commencent par b
Il y a 32143 mots qui commencent par c
...
```

5) Déterminer les fréquences respectives des lettres `a`, `b`, etc. dans l'ensemble des mots du fichier.

```
Fréquence de la lettre a : 9.77990547883021 %
Fréquence de la lettre b : 1.3818917044723495 %
Fréquence de la lettre c : 3.3681792343680037 %
...
```

6) Déterminer les proportions de mots qui contiennent un `a`, un `b`, etc.

```
71.18961905598134 % des mots contiennent un a
13.423353361012506 % des mots contiennent un b
30.39555711651594 % des mots contiennent un c
...
```

7) Quels sont les mots qui contiennent un `q` non suivi d'un `u` ?

8) Quels sont les mots qui contiennent toutes les voyelles (exemple : asymptotique) ?

9) Les mots `defense`, `somnoler`, `majestueux`, `narghile` ont une particularité : ils contiennent trois lettres consécutives (respectivement `def`, `mno`, `stu`, `ghi`). Trouver les mots qui contiennent quatre lettres consécutives.

10) Écrire une fonction `mots_finissant_par` qui, recevant une chaîne de caractères, renvoie la liste des mots qui finissent par cette chaîne.

```
>>> mots_finissant_par('thon')
['berthon', 'marathon', 'python', 'thon', 'zython']
```

11) Écrire une fonction `completer` qui, recevant une chaîne de caractères formée de lettres minuscules et d'astérisques, affiche les mots pouvant compléter cette chaîne (très utile pour les cruciverbistes).

```
>>> completer('p*th**')
pathos
pythie
python
>>> completer('z**************')
zigouillassions
zigouilleraient
zinzinulassions
zinzinuleraient
zootechnicienne
```

12) Écrire une fonction `voisins` qui, recevant une chaîne de caractères `mot`, renvoie la liste des mots qui diffèrent de `mot` d'une et une seule lettre.

```
>>> voisins('liste')
['ciste', 'leste', 'lifte', 'lisse', 'lista', 'piste']
>>> voisins('chaine')
['chaina', 'chaire', 'chaise']
```