

I. Générer toutes les sous-listes d'une liste

Écrire une fonction qui renvoie une liste contenant toutes les sous-listes possibles d'une liste. Par exemple, les sous-listes de [1, 2,3] sont [], [1], [2], [3], [1, 2], [2, 3] et [1, 2, 3]. Noter que votre fonction renverra toujours une liste contenant au moins la liste vide, car la liste vide est une sous-liste de toutes les listes. Tester la fonction sur un exemple.

II. Score au scrabble

Dans le jeu de Scrabble, chaque lettre est associée à des points. Le score total d'un mot est la somme des scores de ses lettres. Les lettres les plus courantes valent moins de points, tandis que les lettres les moins courantes valent plus de points. Les points associés (version française du jeu) à chaque lettre sont indiqués ci-dessous :

- A,E,I,L,N,O,R,S,T,U : **1 point**
- D,G,M : **2 points**
- B,C,P : **3 points**
- F,H,V : **4 points**
- J,Q : **8 points**
- K,W,X,Y,Z : **10 points**

Écrire un programme qui calcule et affiche le score Scrabble d'un mot. Créer un dictionnaire qui établit des correspondances entre les lettres et les valeurs des points. Utiliser ensuite le dictionnaire pour calculer le score.

III. Tokeniser ou découper une chaîne de caractères en jetons

La tokenisation consiste à convertir une chaîne de caractères en une liste de sous-chaînes, appelées jetons. Dans de nombreuses circonstances, une liste de jetons est beaucoup plus facile à utiliser que la chaîne de caractères originale, car cette dernière peut présenter un espacement irrégulier. Dans certains cas, un travail important est également nécessaire pour déterminer où un jeton se termine et où le suivant commence.

Dans une expression mathématique, les jetons sont des éléments tels que les opérateurs, les nombres et les parenthèses. Les symboles des opérateurs que nous examinerons dans

ce problème sont *, /, ^, - et +. Les opérateurs et les parenthèses sont faciles à identifier car le symbole est toujours un caractère unique et le caractère ne fait jamais partie d'un autre symbole. Les nombres sont un peu plus difficiles à identifier, car le symbole peut être composé de plusieurs caractères. Toute séquence de chiffres consécutifs doit être considérée comme un seul jeton numérique.

Écrire une fonction qui prend une chaîne de caractères contenant une expression mathématique comme seul paramètre et la décompose en une liste de jetons. Chaque jeton doit être une parenthèse, un opérateur ou un nombre. (Pour des raisons de simplicité, nous ne travaillerons qu'avec des nombres entiers dans ce problème). Retourner la liste des jetons comme seul résultat de la fonction.

Vous pouvez supposer que la chaîne transmise à votre fonction contient toujours une expression mathématique valide composée de parenthèses, d'opérateurs et d'entiers. Cependant, votre fonction doit gérer des quantités variables d'espaces blancs (y compris l'absence d'espaces blancs) entre ces éléments. Inclure un exemple pour tester votre fonction.

```
expression = "(123/540)*230"
tokenList(expression)
```

votre programme devrait alors s'afficher :

```
[ '(', '123', '/', '540', ')', '*', '230' ]
```

IV. Crible d'Ératosthène

Le crible d'Ératosthène est une technique mise au point il y a plus de 2 000 ans pour trouver facilement tous les nombres premiers compris entre 2 et une certaine limite. Voici une description de l'algorithme :

- Lister tous les nombres de 0 à la limite.
- Barrer 0 et 1 car ils ne sont pas premiers
- Fixer p à 2
- **Tant que** p est inférieur à limite
 - Barrer tous les multiples de p (mais pas lui-même)
 - Fixer p au nombre suivant de la liste qui n'a pas été barré
- Afficher tous les nombres qui n'ont pas été barrés comme premiers

Listing 1: une proposition de début de code.

```
1 limite = int(input("Identifier tous les nombres premiers jusqu'à quelle ↵
    limite ?"))
```
