
IngeSUP - TD 05 - Listes et chaînes 2

"Nous ferons mieux de nous mettre plus haut sur notre propre liste de choses à faire."

Michelle Obama

Exercice 05.1

Pour chacune des questions suivantes, écrire un programme qui remplit une liste d'entiers demandés à l'utilisateur à l'aide d'une boucle et de la fonction input), puis effectuez les actions suivantes :

1. Complétez le programme pour afficher la liste en colonne. C'est-à-dire que chaque élément est du type `i : liste[i]` si `i` est l'indice et `liste` le nom de la liste.

Entrée []:



```
# VOTRE CODE ICI
```

1. Modifiez le programme pour qu'il compte le nombre de multiples de 3 dans la liste

Entrée []:



```
# VOTRE CODE ICI
```

2. Modifiez le programme pour qu'il calcule la somme de toutes les valeurs paires issues de la liste

Entrée []:



```
# VOTRE CODE ICI
```

3. Modifiez le programme pour qu'il indique si la valeur de la moyenne arithmétique des entiers entrés par l'utilisateur est supérieure à 10.

Entrée []:



```
# VOTRE CODE ICI
```

4. Modifiez le programme précédent pour qu'il remplisse une nouvelle liste avec toutes les valeurs comprises dans l'intervalle [50,70].

Entrée []:



```
# VOTRE CODE ICI
```

5. Modifiez le programme précédent pour qu'il affiche la liste à l'envers, sans sauter la ligne et sans faire appel à une autre liste.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 05.2

1. Ecrire un programme qui calcule et stocke dans une liste `liste2` le double des éléments contenus dans une liste `liste` en utilisant la syntaxe des compréhensions de liste.

Entrée []:



```
# VOTRE CODE ICI
```

2. Ecrire un programme qui stocke dans une liste `L2` les éléments impairs d'une liste `L` en partant du début de la liste en utilisant la syntaxe des compréhensions de liste.

Entrée []:



```
# VOTRE CODE ICI
```

3. Ecrire un programme qui remplisse une liste `L2` avec des nombres pairs parmi les 5 premières valeurs d'une liste `L` en utilisant la syntaxe des compréhensions de liste. Afficher `L2`.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 05.3

Ecrire un programme qui découpe, dans une liste `mots`, une chaîne de caractères en mot (c'est à dire une chaîne séparée par des espaces) et renvoie le nombre de caractères dans une seconde liste `nb_caracteres` dans le même ordre que la liste des mots `mots`.

Remarques :

1. On fera bon usage de la fonction `split()`
2. On pourra utiliser la syntaxe des compréhensions de liste
3. La liste `mots` est une liste qui peut servir d'intermédiaire

Entrée []:



```
# VOTRE CODE ICI
```

Entrée []:



```
#version compréhensions de liste
```

Exercice 05.4

Ecrire un programme qui génère les chaînes `tata`, `tete`, `titi`, `toto` et `tutu` à partir de la chaîne `t_t_` ou n'importe quelle chaîne composée de consonnes et du caractère `_` qui sera remplacé par toutes les voyelles `aeiou`. L'ensemble des variantes ainsi construite sera regroupé dans une liste.

Remarques :

1. On fera bon usage de la fonction ``replace``
2. On pourra utiliser la syntaxe des compréhensions de liste

Entrée []:



```
#VOTRE CODE ICI
```

Entrée []:



```
# version compréhension de Liste
```

Exercice 05.6

Ecrire un programme qui affecte la valeur `True` à une variable *toute* si une liste *liste* possède des valeurs toutes supérieures à 1.

Entrée []:



```
# VOTRE CODE ICI
```

Exercice 05.7

1. Ecrire un programme qui effectue une permutation circulaire de la liste. Chaque élément prend la valeur de son élément de droite et la première valeur se retrouve en dernier : Par exemple si la liste est [8, 24, 48, 2, 16] après l'exécution du programme elle sera [24, 48, 2, 16, 8]

Entrée []:



```
# VOTRE CODE ICI
```

2. Ecrire un programme qui modifie une liste par miroir, c'est-à-dire que tout élément situé à la place *i* en partant du début est échangé avec son correspondant à la place *i* en partant de la fin. Par exemple si la liste est [8, 24, 48, 2, 16] après l'exécution du programme elle sera [16, 2, 48, 24, 8].

Entrée []:



```
# VOTRE CODE ICI
```

3. Dédurre de la question précédente un programme qui détermine si une liste est symétrique.

Indication: Une liste est symétrique si elle est égale à son miroir

Entrée []:



VOTRE CODE ICI

Corrigé du TD 05

Vous pouvez retrouver le corrigé de ce TD [ici \(Corrig%C3%A9s/Corrig%C3%A9_TD%2004.ipynb\)](#).