

TP1**Compelxité**

1. Écrire la fonction somme factorielles permettant de calculer la somme suivant :

$$\text{somme_factorielles}(n) = \sum_{k=0}^n k!$$

```
>>> somme_factorielles(0)
>>> 1
>>> somme_factorielles(3)
>>> 10
```

2. Écrire un programme permettant de trouver le premier élément qui se répète dans une liste d'entiers.

```
>>> l = [1, 2, 3, 5, 7, 5, 2, 12, 1]
>>> premiere_repetition(l)
>>> 5
```

3. Étant donné la matrice bidimensionnelle, écrivez un algorithme pour imprimer toutes les diagonales de la matrice. Exemple : Pour la matrice suivante :

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

Le résultat est suivant :

```
1
5 2
9 6 3
13 10 7 4
14 11 8
15 12
16
```

4. Etant donné une liste d'entiers dont chacun apparaît deux fois excepté un qui apparaît une seule fois. Écrire une fonction permettant de trouver ce nombre. Donnez sa complexité et proposer un algorithme de complexité linéaire.

```
>>> l = [1, 2, 3, 5, 7, 5, 2, 1, 3]
>>> intrus(l)
>>> 7
```

5. Etant donné une liste d'entiers non triés, écrire un programme permettant de tester si la liste contient tous les entiers d'un certain intervalle. Donnez sa complexité et proposer un algorithme de complexité linéaire.

```

>>> l = [4,1,2,3,5,7,5,2,1,3,6,9]
>>> intervalle_est_dans_liste(1,1,12)
>>> False
>>> intervalle_est_dans_liste(1,2,5)
>>> True

```

6. Etant donnée une liste l . Écrire un programme permettant de trouver une instance de i, j, k tel que $0 \leq i < j < k < n$ et $l[i] < l[j] < l[k]$.

```

>>> l = [4,0,7,3,5,7,5,2,1,3,6,9]
>>> triplet_avec_ordre(l)
>>> (1,3,4)
>>> l = [14,12,7,5,5,3,0]
>>> triplet_avec_ordre(l)
>>> (-1,-1,-1)

```

— Donnez sa complexité et proposer un algorithme de complexité linéaire.