

DS 1

le 26/09/2026 – Durée : 2h

- L'usage de la calculatrice n'est pas autorisé.
- La clarté et la précision des raisonnements interviendront pour une grande part dans la notation. La présentation (en particulier les indentations) est également essentielle.
- Le résultat d'une question peut être admis afin de traiter une question suivante ; une fonction peut être utilisée dans la suite d'un exercice même si elle n'a pas été écrite.
- Le barème tiendra compte de la longueur du devoir.

Exercice 1

1. Écrire une fonction `factorielle` qui prend en argument un entier naturel n et renvoie $n!$. (On n'acceptera pas bien sûr de réponse utilisant la fonction `factorial` du module `math` de python).
2. Écrire une fonction `seuil` qui prend en argument un entier M et renvoie le plus petit n tel que $n! > M$. On demande une complexité linéaire en l'entier n tel que $(n-1)! \leq M < n!$, la multiplication des entiers étant considérée comme une opération élémentaire.
3. Écrire une fonction booléenne `est_divisible` qui prend en argument un entier n et renvoie `True` si $n!$ est divisible par $n+1$ et `False` sinon.
4. On considère la fonction suivante nommée `mystere`, définie pour $n \in \mathbb{N}^*$:

```
def mystere(n):  
    s=0  
    for k in range(1,n+1):  
        s=s+factorielle(k)  
    return s
```

- (a) Quelle valeur renvoie `mystere(4)` ?
 - (b) Déterminer le nombre de multiplication qu'effectue `mystere(n)`.
 - (c) Proposer une amélioration du script de la fonction `mystere` afin d'obtenir une complexité linéaire.
5. (a) Écrire une fonction récursive `mystere_rec(n)` qui renvoie le même résultat que la fonction `mystere(n)` (on s'autorise à utiliser la fonction `factorielle`).
Quelle est la complexité de cette fonction ? Justifier.
- (b) De la même manière que dans la question 4., on cherche à obtenir une complexité linéaire, tout en conservant le caractère récursif de la fonction.
Écrire une fonction récursive `mystere2_rec(n)` qui renvoie une liste dont le premier élément est $n!$ et le second le résultat de `mystere(n)`. On n'utilisera pas la fonction `factorielle` et on demande une complexité linéaire (que l'on justifiera).

Problème

Vous traiterez le problème suivant à l'exception des questions 5 à 9 (incluses).

Remarque : les parties de ce problème sont largement indépendantes.