

Dans ce TD, “trié” signifie “trié par ordre croissant”.

*** Les exercices marqués d’une étoile sont à faire à la maison.**

Algorithm 1 Algorithme PUISS

Entrée : n un entier naturel.

```
1: fonction PUISS( $k, n$ )
2:   si  $n = 0$  alors
3:     retourne 1
4:   sinon
5:     retourne ( $k \cdot \text{PUISS}(n - 1)$ )
```

Algorithm 2 Algorithme PUISSAUX

Entrée : n , et a deux entiers naturels.

```
1: fonction PUISSAUX( $k, n, a$ )
2:   si  $n = 0$  alors
3:     retourne  $a$ 
4:   sinon
5:     retourne PUISSAUX( $k, (n - 1), (a * k)$ )
```

Exercice 1. *Le Terminal.*

1. Comment réutiliser PUISSAUX pour créer un algorithme équivalent à PUISS (on appellera cet algorithme PUISSTER) ?
2. Calculer la main, PUISS(5), et PUISSTER(5), en suivant rigoureusement les instructions ? Que constatez-vous ?
3. En vous inspirant de la question 1, et de PUISSAUX, adaptez l'algorithme SOMME afin de le rendre moins gourmand en mémoire.

Algorithm 3 Algorithme SOMME

Entrée : T Un tableau de taille t , n un entier.

```
1: fonction SOMME( $n$ )
2:   si  $n = t$  alors
3:     retourne 0
4:   sinon
5:     retourne ( $T[n] + \text{SOMME}(n + 1)$ )
```

Exercice 2. *Comptage.*

1. Ecrire un algorithme prenant en entrée un tableau, un élément x , un entier n , qui compte le nombre d'occurrence de x dans le sous-tableau $T[0, (n - 1)]$. On l'appellera COMPTAGEAUX.

2. Utiliser cet algorithme pour construire un algorithme qui compte le nombre d'occurrence d'un élément x dans un tableau T (On l'appellera COMPTAGE).
3. Prouver par récurrence la correction de COMPTAGEAUX, en déduire la correction de COMPTAGE.

Exercice 3. *Construction d'une liste.*

1. Quelle liste est stockée dans L après la suite d'instructions suivante ? (Faites un dessin.)

```

1  a := new Cellule(1, nil)
2  b := new Cellule(2, a)
3  c := new Cellule(3, nil)
4  a.next := c
5  b.val := 4
6  L := new List(b)

```

2. On suppose que la liste M contient la suite de valeurs (1, 2, 3, 4, 5). Écrivez la suite d'instructions qui mute M pour qu'elle contienne (1, 2, 4, 5) sans créer aucune nouvelle cellule et sans jamais modifier la valeur contenue dans une cellule.
3. On suppose maintenant que la liste N contient la suite de valeurs (1, 2, 4, 5). Écrivez la suite d'instructions qui mute N pour qu'elle contienne (1, 2, 3, 4, 5), en créant une seule nouvelle cellule.

Exercice 4. *Manipulation de listes.*

1. Écrivez un algorithme qui prend en entrée une liste L non vide et retourne son élément maximal.
2. Écrivez un algorithme qui prend en entrée une liste L et retourne 1 si L est triée, et 0 sinon (la liste vide est triée).

Exercice 5. *Palindrome.*

* Ecrire un algorithme récursif qui prend en entrée une chaîne de caractères (que l'on pourra considérer comme un tableau de caractères) qui indique si une chaîne de caractères est un palindrome. On hésitera pas à se servir d'un algorithme auxiliaire.