

# Licence 3 Informatique

## Méthodes formelles d'aide à la détection d'erreurs

Il vous est demandé un travail personnel. Toute réponse non prouvée sera considérée comme fausse. En clair, nous voulons une réponse, l'explication qui conduit à formuler cette réponse et la preuve que la réponse est valide.

Dans ce devoir, TabEl sera un tableau d'éléments indicé de 0 à n.

$\leq$  une relation d'ordre totale sur les éléments du tableau (< désigne la version stricte de cet ordre)

### Exercice 1 :

Soit PremierCalcul la fonction suivante

**Fonction** PremierCalcul

**Données** TabEl : un tableau d'éléments indicé de 0 à n ;

i : un indice de TabEl // Précondition :  $0 \leq i \leq n$

**Résultat** un entier naturel

**Variables** j, res : entier naturel

**DébutCode**

j  $\leftarrow$  0 ; res  $\leftarrow$  0

**TantQue** j  $\leq$  n **faire**

**Si** TabEl[j] < TabEl[i] ou (TabEl[j] = TabEl[i] et j < i) **alors** res  $\leftarrow$  res+1 **Finsi**

j  $\leftarrow$  j+1 ;

**FinTq**

**Renvoyer** (res)

**FinCode**

Question 1.1 : En utilisant l'ordre alphabétique pour les comparaisons, réalisez les traces des appels suivants : PremierCalcul (Tableau\_1, 2) et PremierCalcul (Tableau\_1, 4)<sup>1</sup>.

Question 1.2 : Démontrez que La fonction PremierCalcul s'arrête pour toute donnée.

Question 1.3 : Donnez la complexité en temps de cet algorithme

---

<sup>1</sup> Tableau\_1 se trouve à la fin du sujet

## Exercice 2 :

Soit SecondCalcul la fonction suivante

**Fonction** SecondCalcul

**Donnée**            TabEl : un tableau d'éléments indicé de 0 à n ;

**Résultat**        un tableau d'entier indicé de 0 à n ;

**Variables**        j : entier naturel

                      TabVal : un tableau d'entier indicé de 0 à n

**DébutCode**

    j ← 0 ;

**TantQue** j ≤ n **faire** TabVal[j] ← PremierCalcul (TabEl, j) ; j ← j+1 **FinTq**

**Renvoyer** (TabVal)

**FinCode**

Question 2.1 : En utilisant l'ordre alphabétique pour les comparaisons, réalisez les traces de l'appel suivant : SecondCalcul (Tableau\_1).

Question 2.2 : Démontrez que La fonction SecondCalcul s'arrête pour toute donnée.

Question 2.3 : Donnez la complexité en temps de cet algorithme

Pour la suite nous utiliserons l'algorithme d'échange suivant

**Algorithme** Echange

**Donnée/Résultat** T : Un tableau de valeurs

**Données** i, j : deux indices de T

**Variable** inter : valeur

**DébutCode**

    inter ← T[i] ; T[i] ← T[j] ; T[j] ← Inter

**FinCode**

### Exercice 3 :

Soit TroisièmeCalcul l'algorithme suivant :

**Algorithme** TroisièmeCalcul

**Données/Résultats**      TabEl : un tableau d'éléments indicé de 0 à n ;

                                 TabVal : un tableau d'entier indicé de 0 à n

// **Précondition** : TabVal contient une permutation<sup>2</sup> des entiers naturels compris entre 0 et n ([0..n])

**Variable** j : entier naturel

**DébutCode**

    j ← 0 ;

**TantQue** j ≤ n **faire**

**Si** TabVal[j] = j **alors** j ← j+1

**Sinon** Echange (TabEl, j, TabVal[j]) ; Echange (TabVal, j, TabVal[j]) ;

**FinSi**

**FinTq**

**FinCode**

Question 3.1 : Calculez les traces des appels suivants :

TroisièmeCalcul (Tableau\_1, Tableau\_2) et TroisièmeCalcul (Tableau\_1, Tableau\_3).

Question 3.2 : L'algorithme TroisièmeCalcul s'arrête-t-il pour toute donnée ?

Question 3.3 : Quelle est la complexité de l'algorithme TroisièmeCalcul lorsqu'il s'arrête ?

---

<sup>2</sup> Voir <https://fr.wikipedia.org/wiki/Permutation> ou <https://www.techno-science.net/definition/6069.html> pour avoir une définition du concept.

#### Exercice 4 :

Soit l'algorithme Mystère suivant :

#### **Algorithme Mystère**

**Donnée/Résultat**      TabEl : un tableau d'éléments indicé de 0 à n ;

**Variables**              j : entier naturel

TabVal : un tableau d'entier indicé de 0 à n

#### **DébutCode**

TabVal  $\leftarrow$  SecondCalcul(TabEl) ;

TroisièmeCalcul (TabEl, TabVal)

#### **FinCode**

Question 4.1 : En utilisant l'ordre alphabétique pour les comparaisons, calculez la trace de l'appel suivant : Mystère (Tableau\_1).

Question 4.2 : Démontrez que Mystère, lorsqu'il s'arrête, est un algorithme qui trie les éléments de TabEl. Pour faire cette preuve vous écrirez et prouverez les spécifications de PremierCalcul, SecondCalcul et TroisièmeCalcul.

Question 4.3 : L'algorithme Mystère s'arrête-t-il pour toute donnée ?

Question 4.4 : Quelle est la complexité de l'algorithme Mystère lorsqu'il s'arrête ?

Données pour les traces :

0	1	2	3	4	5	6	7	8	9
France	Chine	Népal	Chili	Chine	USA	Angola	Turquie	Chine	Pérou

Tableau\_1 : Un exemple de tableau

0	1	2	3	4	5	6	7	8	9
4	3	5	2	6	1	7	0	9	8

Tableau\_2 : Un exemple de tableau

0	1	2	3	4	5	6	7	8	9
4	3	5	2	6	1	8	0	9	8

Tableau\_3 : Un exemple de tableau