

## Méthodologie de la Programmation

1ère année Apprentissage  
Décembre 2022.  
Durée 1h30

Tous les programmes demandés seront écrits en langage algorithmique.  
Ces programmes devront respecter scrupuleusement TOUTES les consignes de bonne programmation définies en cours, TD et TP

### Exercice 1. Recherche d'élément majoritaire (7 points)

Cet exercice s'intéresse à la recherche d'un élément majoritaire dans un tableau d'entiers.

On considère un tableau  $T[1..Nmax]$  d'entiers contenant  $N \leq Nmax$  entiers.

Un élément  $x$  de  $T[1..N]$  est un élément majoritaire si et seulement si le nombre d'occurrences de  $x$  dans  $T[1..N]$  est strictement plus grand que  $N/2$  où  $/$  est la division entière. Le nombre d'occurrences d'un élément  $x$  dans un tableau  $T[1..N]$  désigne le nombre de fois où  $x$  apparaît dans  $T[1..N]$ .

Un élément majoritaire n'existe pas forcément, mais s'il existe, alors il est unique.

Exemple 1 : pour le tableau ci-dessous avec  $N = 10$  et  $Nmax = 12$

6	1	6	6	3	6	8	6	6	7		
---	---	---	---	---	---	---	---	---	---	--	--

l'élément majoritaire est 6.

Exemple 2 : pour le tableau ci-dessous avec  $N = 10$  et  $Nmax = 12$

6	1	6	7	3	5	8	4	2	7		
---	---	---	---	---	---	---	---	---	---	--	--

il n'y a pas d'élément majoritaire.

#### Questions

- Q1.1 Déclarer le type  $T\_Tab$  correspondant au tableau ci-dessus.
- Q1.2 Donner en langage algorithmique le contrat et l'entête d'un sous-programme *Majoritaire* qui calcule l'élément majoritaire dans un tableau s'il existe. Le sous-programme *Majoritaire* signalera aussi que l'élément majoritaire existe ou n'existe pas.
- Q1.3 Raffiner le sous-programme *Majoritaire* en langage algorithmique, de manière itérative, depuis le niveau  $R_0$  jusqu'au niveau  $R_n$  correspondant à l'algorithme final.
- Q1.4 Donner l'algorithme final obtenu

### Exercice 2. Codage de De Vigénère (6 points)

Dans cet exercice, on manipule des chaînes de caractères de type  $T\_CHAINE$  défini ci-dessous.

```
----  
CONSTANTE entier Longueur_Max <-- 100  
----  
TYPE T_CHAINE est TABLEAU(1..Longueur_Max) de Caractère
```

Nous nous proposons d'étudier une technique de cryptage de texte par décalage.

Le principe du cryptage consiste à produire, en résultat d'un algorithme, une chaîne de caractères correspondant au cryptage d'une autre chaîne de caractères, en entrée de cet algorithme. Ce cryptage est réalisé à l'aide d'une clé qui est elle-même une chaîne de caractères dans le cas du codage de De Vigenère. La chaîne de caractères en entrée peut représenter un mot à coder ou une phrase à coder (dans ce dernier cas les mots de la phrase sont séparés par un caractère ' ' c'est-à-dire un espace).

Pour le codage de De Vigenère, abordé dans cet exercice, on fait les hypothèses suivantes :

1. la chaîne de caractères à coder ne comporte que des lettres en majuscules sans accent. Aucun caractère spécial comme l'apostrophe ou le tiret ne fera partie des phrases étudiées. Seul, le caractère ' ' (espace) sera utilisé : il permet de séparer les mots d'une phrase. Le caractère ' ' (espace) est codé par lui-même (et n'entre donc pas en considération).
2. les caractères alphabétiques sont indexés de 0 pour le caractère 'A' à 25 pour le caractère 'Z'. On pourra construire un tableau d'entier `code`, indexé par les caractères dont les valeurs correspondent aux différents codes des lettres. Par exemple, `code('B')` vaut 1.

Dans le codage de De Vigenère, les décalages utilisés dépendent d'une clé qui est une chaîne de caractères.

Par exemple, si la clé est "BONJOUR", les lettres du message seront décalées de :

```
1 lettre ('B' a pour code 1)  
14 lettres ('O' a pour code 14)  
13 lettres ('N' a pour code 13)  
9 lettres ('J' a pour code 9)  
14 lettres ('O' a pour code 14)  
20 lettres ('U' a pour code 20)  
17 lettres ('R' a pour code 17)
```

Ainsi pour coder le message "ELEVE DE PREMIERE ANNEE", avec la clé "BONJOUR" on répète la clé autant de fois que nécessaire et procède donc comme suit :

```
ELEVE DE PREMIERE ANNEE  
+ BONJO UR BONJOURB ONJOU
```

```
-----  
FZRES XVQFR VWYIF OAWSY
```

Le codage de la même phrase avec une autre clé 'IMPERATIVE' donne :

```
ELEVE DE PREMIERE ANNEE  
+ IMPER AT IVEIMPER ATIVE
```

```
-----  
MXTZV DXXMI UUTVV AGVZI
```

Questions

- Q2.1 Donner le contrat d'un sous-programme de codage de De Vigenère appelé **Vigenere**.  
Q2.2 Raffiner cette spécification pour obtenir un algorithme qui implante cette spécification.

Exercice 3. Le drapeau hollandais (7 points)

L'algorithme du drapeau hollandais a été proposé par E. Dijkstra, il est à la base de l'algorithme de tri appelé quicksort.

Considérons un tableau  $T[1..Nmax]$  d'entiers contenant  $N \leq Nmax$  entiers, et une valeur particulière  $V$  appartenant à  $T[1..N]$  appelée pivot.

On dit qu'un tableau satisfait la condition du drapeau hollandais s'il peut être mis sous la forme suivante :

Valeurs de $T[1..N]$ strictement inférieures à $V$ non nécessairement triées	Occurrences de $V$	Valeurs de $T[1..N]$ strictement supérieures à $V$ non nécessairement triées
---	--------------------	---

Par exemple pour le tableau ci-dessous avec  $N = 10$ ,  $Nmax = 12$

6	1	9	8	3	6	8	9	5	7		
---	---	---	---	---	---	---	---	---	---	--	--

et une valeur  $V = 6$ , l'algorithme du drapeau Hollandais donne un tableau de la forme suivante appelé tableau Hollandais :

5	1	3	6	6	8	9	8	9	7		
---	---	---	---	---	---	---	---	---	---	--	--

L'objectif de cet exercice est de mettre un tableau sous la forme d'un tableau Hollandais. Pour résoudre ce problème, on utilisera un seul tableau.

Questions

- Q3.1 Déclarer le type  $T\_Tab$  correspondant au tableau ci-dessus.  
Q3.2 Donner en langage algorithmique le contrat et l'entête d'un sous-programme *Drapeau* qui renvoie le tableau drapeau hollandais d'un tableau donné pour une valeur pivot donnée  
Q3.3 Raffiner le sous-programme *Drapeau* en langage algorithmique, de manière itérative, depuis le niveau  $R_0$  jusqu'au niveau  $R_n$  correspondant à l'algorithme final.  
Q3.4 Donner l'algorithme final obtenu