Méthodologie de la Programmation

1ère année Apprentissage
Décembre 2019.
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**

.

1 Questions de cours

C.1. Y a-t-il des liens entre méthode des raffinages et sous-programmes ?

C.2. Y a-t-il des liens entre raffinages et architecture logicielle ?

C.3. Qu'est ce qu'un module générique ? Quel est son intérêt ?

C.4. A quoi sert une précondition ? Une postcondition ?

C.5. Comment différenciez-vous programmation offensive et programmation défensive ?

Exercice 1. Recherche d'élément majoritaire

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.

Exercice 2. Vérification d'un sudoku

Dans cet exercice, nous souhaitons écrire un algorithme qui vérifie si un sudoku de 9 éléments est correct ou pas.

Un sudoku de neuf éléments est composé d'un tableau à deux dimensions $Sudoku[1..9, 1..9]$ dont chaque case contient un chiffre compris entre 1 et 9.

Un sudoku est correct si et seulement si :

- les 9 lignes contiennent une seule occurrence de chaque chiffre compris entre 1 et 9
- les 9 colonnes contiennent une seule occurrence de chaque chiffre compris entre 1 et 9
- les 9(3×3) sous-tableaux contiennent une seule occurrence de chaque chiffre compris entre 1 et 9

Exemple de sudoku correct

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

Questions

Q2.1 Définir le type de données T_Sudoku représentant un sudoku.

Q2.2 Donner en langage algorithmique le contrat et l'entête d'un sous-programme $Sudoku\_OK$ qui indique si un sudoku est correct ou non.

Q2.3 Raffiner le sous-programme $Sudoku\_OK$ en langage algorithmique, de manière itérative, depuis le niveau R_0 jusqu'au niveau R_n correspondant à l'algorithme final.

Exercice 3. Le drapeau hollandais

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.

Q1.2 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

Q1.3 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.