

Algorithmique des tableaux

Le sujet comporte 7 pages. Aucun document n'est autorisé. Répondez directement sur le sujet.

Exercice 1 Les booléens

Pour chaque valeur des variables **x** et **y** complétez le tableau suivant :

- en calculant la valeur de l'expression correspondante et
- en précisant les étapes de calcul nécessaires pour obtenir le résultat.

	Expression	Valeur	Justification
x=25, y=3	$\text{not}(x\%5==0 \text{ and } y<x)$		
x=5, y=0	$x\%2==1 \text{ or } x//y>=1$		
x=0, y=1	$(x!=0 \text{ or } x!=1) \text{ and } (y!=0 \text{ or } y!=1)$		

Exercice 2 Ecriture d'algorithme

On considère un tableau *t* (non trié) d'entiers représentant les années de naissance des profs d'algorithmique des tableaux (anonyme, bien sûr!).

1. Ecrivez une fonction `fete_40_ans(t,n)` qui teste (retourne `True` si vrai et `False` sinon) si l'un des profs fête ses 40 ans en 2018.

2. Ecrivez une fonction `tous_vieux(t,n)` qui teste si tous les profs ont plus de 30 ans (c'est à dire ont au moins 31 ans courant 2018).

3. Ecrivez une fonction `les_annees_en_8(t,n)` qui *affiche* les années de naissance des profs qui fêtent un age multiple de 10 en 2018.

4. Ecrivez une fonction `le_plus_jeune(t,n)` qui retourne l'année de naissance du plus jeune prof. (Aide : entre une personne née en 1985 et une personne née en 1960 laquelle est la plus jeune?)

5. Ecrivez une fonction `nombre_de_quinqua(t,n)` qui retourne le nombre de quinquagénaires (c'est à dire qui fêtent un age entre 50 et 59 ans en 2018).

Exercice 3 Lecture d'algorithme

Considérons la fonction `mystere` suivante prenant en paramètre un tableau `t` contenant `n` nombres et un entier `k` avec $0 \leq k < n$:

```
def mystere(t, n, k):
    for i in range(k):
        for j in range(k,n):
            if t[i] > t[j]:
                return False
    return True
```

1. Soit `t=[2,1,9,5]`, simulez l'exécution de l'appel `mystere(t,4,2)` en complétant le tableau suivant montrant l'évolution des variables `n`, `k`, `i` et `j` :

<i>n</i>		
<i>k</i>		
<i>i</i>		
<i>j</i>		

2. Si `t=[2,1,9,5]`, que retourne l'appel `mystere(t,4,2)` ?

3. Si `t= [2,8,4,9,5]`, que retourne l'appel `mystere(t,5,3)` ?

4. Que fait la fonction `mystere` en général ?

5. Déterminer, en fonction de n et k , le nombre de comparaisons effectuées par `mystere`.

--

6. Il existe un algorithme permettant d'écrire une version de la fonction `mystere` qui n'effectue que $n-1$ comparaisons : pouvez vous le décrire ? (il n'est pas nécessaire de donner le code Python)

--

Exercice 4 Suppression d'éléments consécutifs dans un tableau.

Rappel : dans un tableau contenant n éléments les indices $> n-1$ et le contenu des cases associées à ces indices dans le tableau ne font plus partie des données utiles, ils ne seront donc ni considérés dans les algorithmes, ni représentés dans les exemples.

1. Écrire une fonction `supprimer(t, n, k)` qui supprime l'élément situé à la position k du tableau t contenant n éléments en supposant $0 \leq k < n$. L'ordre initial des éléments du tableau est conservé. La fonction retourne le nouveau nombre d'éléments.

Exemple :

Après appel de la fonction `supprimer(tab, 8, 3)`, le tableau

`tab=`

0	1	2	3	4	5	6	7		
1	2	43	7	29	6	9	100		

 où la 1^{ère} ligne représente les indices

devient

0	1	2	3	4	5	6			
1	2	43	29	6	9	100			

 et la valeur retournée est 7

--

2. En utilisant la fonction `supprimer`, écrire une fonction `supprimerEntre(t, n, p, q)` qui supprime du tableau `t` à `n` éléments tous les éléments situés entre les positions `p` et `q` comprises. On suppose $0 \leq p \leq q < n$. L'ordre initial des éléments du tableau est conservé. La fonction renvoie le nouveau nombre d'éléments.

Exemple :

Après appel de la fonction `supprimerEntre(tab, 8, 3, 5)`, le tableau

tab=	0	1	2	3	4	5	6	7		
	1	2	43	7	29	6	9	100		

devient	0	1	2	3	4					
	1	2	43	9	100					

et la valeur retournée est 5

3. Ecrire une seconde version de la fonction `supprimerEntre(t, n, p, q)`, cette fois sans utiliser la fonction `supprimer`. Il faut placer directement à leur position définitive tous les éléments du tableau qui doivent être déplacés.

Exercice 5 Système monétaire

Un *système monétaire* S est un tableau d'entiers (représentant des pièces en euros) rangés par ordre strictement croissant dont le premier élément est 1.

Par exemple le système monétaire représenté par le tableau suivant : $S = [1, 2, 5, 10, 50]$ est composé de cinq types de pièces différentes : les pièces de 1 €, 2 €, 5 €, 10 € et 50 €.

On souhaite qu'un automate distributeur d'argent paye un montant entier $m \geq 1$ à l'aide d'un système monétaire S en respectant la règle suivante : "Utiliser autant de fois que possible la pièce de plus forte valeur de S inférieure ou égale au montant, en recommençant éventuellement avec ce qu'il reste à payer, et ce jusqu'à atteindre m exactement".

Dit autrement, l'automate doit payer en utilisant les pièces de valeurs les plus fortes d'abord.

Par exemple pour le système monétaire S précédant et pour $m=36$, le distributeur devra payer avec trois pièces de 10 €, une pièce de 5 € et enfin une pièce d'1 €.

On supposera que le distributeur possède suffisamment de pièces de chaque valeur pour payer le montant souhaité. On s'intéresse au nombre de pièces rendues par l'automate. Dans l'exemple précédant, le nombre total de pièces rendues était de 5.

1. Avec le système monétaire S et $m = 24$ combien de pièces l'automate rendra-t-il et quelles sont-elles ?

2. Écrivez une fonction `nbPieces(m,S)` qui renvoie le nombre de pièces rendues par un automate respectant la règle ci-dessus pour payer un montant m dans un système monétaire S .

Nom:

Prénom:

Groupe:

