

TEST 1

Nom :

Prénom :

Exercice 1 :

Soit la fonction `mystere` qui prend en paramètres un tableau `c` contenant un nombre `n` pair d'éléments :

```
1 def mystere(c, n):
2
3     k = 0
4     for i in range(n // 2):      # premiere boucle
5         k += c[2*i]
6         t = [-1]*k
7         j = 0
8
9     for i in range(0, n, 2):      # deuxieme boucle
10        for r in range(c[i]):
11            t[j] = c[i+1]
12            j += 1
13    return t
```

Exemple d'utilisation:

```
mystere([1, 3, 3, 2, 2, 8], 6)
[3, 2, 2, 2, 8, 8]
```

- Remplir le tableau ci-dessous pour la première boucle de la fonction avec l'appel `mystere([3, 0, 1, 9, 2, -2, 5, 0], 8)`

i										
c[2*i]										
k										

- Indiquer le contenu du tableau `t` à la fin de la première itération de la deuxième boucle ($i = 0$).

- Indiquer la valeur suivante de i puis le contenu du tableau `t` à la fin de la deuxième itération.

- Que fait la fonction `mystere` dans le cas général? Préciser dans votre réponse ce que représente `k`.

5. Donner le nombre d'affectations sur **t** en fonction de **k** dans le pire et le meilleur cas.

Exercice 2

Écrire une fonction **sansDoublon(t, n)** qui retourne True si un tableau d'entiers **t** contenant **n** éléments est sans doublon (c'est à dire sans apparition multiple d'un élément), False sinon.

Exemples :

`sansDoublon([1, 3, 3, 5, 0], 5)` renvoie False

`sansDoublon([1, 3, 8, 5, 0], 5)` renvoie True

Quelle est la complexité de votre fonction?