

## TP 3 : boucles et tableaux

L. Naert – T. Ferragut – E. Lemonnier

5 juillet 2024

### Objectifs du TP

- Comprendre le fonctionnement d'un tableau
- Écrire des programmes avec des tableaux

Pour les rendus, il est indispensable de respecter les conventions java et de choisir les cas de test de manière à ce qu'ils soient faciles à comprendre et le plus exhaustif possible. Il sera donc nécessaire d'afficher le contenu des tableaux manipulés. La fonction `displayTab` ci-dessous permet cet affichage. N'hésitez pas à l'utiliser comme dans l'exemple ci-dessous. Remarquez qu'il faut mettre le code de la fonction après le bloc de code de `principal` et non dedans.

```
/**
 * La fonction displayTab(...) permet d'afficher le tableau
 * mis en paramètre (entre parenthèse).
 * @author L. Naert
 */
class Exemple{
    void principal(){
        int[] t = {5,7,0,6,10,8,4,1,35,25,8,3};
        System.out.print("Avant\t    : ");
        displayTab(t);
        t[0] = 12;
        System.out.print("Après\t : ");
        displayTab(t);
    }

    void displayTab(int[] t){
        int i = 0;
        System.out.print("{");
        while(i<t.length-1){
            System.out.print(t[i] + ",");
            i=i+1;
        }
        System.out.println(t[i]+"}");
    }
}
```

## Exercice 1 (reprise du TD3) (\*\*)

Écrire un programme java qui décale d'une case vers la fin le contenu d'un tableau `int[] t` déjà saisi. Le décalage est circulaire dans le sens où la valeur de la dernière case de `t` va en `t[0]`.

Par exemple, le décalage du tableau {5, 7, 0, 6, 10, 8, 4, 1} produit {1, 5, 7, 0, 6, 10, 8, 4}.

**Rendre le code du programme avec une version boucle `while` et une version boucle `for` et les tests d'exécution**

## Exercice 2 (reprise du TD3) (\*\*)

Écrire un programme java qui calcule le cumul des valeurs d'un tableau `int[] t` dans une tableau `int[] cumul` :

- la case `cumul[0]` doit contenir `t[0]`,
- la case `cumul[1]` doit contenir `t[0] + t[1]`,
- la case `cumul[2]` doit contenir `t[0] + t[1] + t[2]`,
- la case `cumul[3]` doit contenir `t[0] + t[1] + t[2] + t[3]`,
- etc

Par exemple, sur le tableau {5, 7, 0, 6, 10, 8, 4, 1} le tableau `cumul` contient {5, 12, 12, 18, 28, 36, 40, 41}.

**Rendre le code du programme avec une version boucle `while` et une version boucle `for` et les tests d'exécution**

## Exercice 3 (\*\*\*)

Écrire un programme java qui détermine la longueur de la plus grande séquence croissante d'un tableau `int[] t` déjà saisi. Par exemple, sur le tableau {5, 7, 0, 6, 10, 8, 4, 1} la plus grande séquence croissante est 3, pour `t[2]`, `t[3]` et `t[4]`.

Le programme doit s'arrêter dès qu'il n'est plus possible de trouver une séquence plus longue.  
**Entre la boucle `for` et `while`, laquelle est la plus adaptée ?**

**Rendre le code du programme et les tests d'exécution**

## Exercice 4 (\*\*)

Modifier le programme précédent pour qu'il affiche également les indices de début et de fin de la plus longue séquence croissante. Par exemple, sur le tableau {5, 7, 0, 6, 10, 8, 4, 1} la plus grande séquence croissante est 3, de l'indice 2 à l'indice 4.

**Rendre le code du programme et les tests d'exécution**

## Exercice 5 (\*\*)

Écrire un programme qui tire aléatoirement 1000 entiers entre 0 et 9 (inclus) et qui stocke dans un tableau le nombre de fois où chaque valeur a été tirée. Afficher ce tableau. Par exemple, `tirageAleatoire = {104,78,...}` si 0 a été tiré 104 fois, 1 78 fois...

**Rendre le code du programme (boucle while ou for au choix) et les tests d'exécution**

## Exercice 6 (\*\*) : Conjecture de Syracuse

Écrire un programme qui demande un entier à l'utilisateur. Le programme doit redemander un entier tant que celui entré par l'utilisateur n'est pas strictement positif. Puis, tant que cet entier n'est pas égal à 1, le programme le divise par deux s'il est pair ou le multiplie par trois et lui ajoute un s'il est impair. Ce programme doit afficher le nombre d'étapes pour arriver à la valeur 1 et le nombre maximal rencontré en chemin. Par exemple, avec 7 comme nombre de départ : le nombre d'étapes doit être de 16 et le nombre max rencontré est 52. **Il y a 2 boucles dans ce programme. Énoncer le rôle de chacune. Pour chacune de ces boucles, quelle syntaxe Java est-il préférable d'utiliser (for/while/do-while) ?**

**Rendre le code du programme et les tests d'exécution**

## Exercice 7 (\*\*) : Exercice facultatif

Étant donné un tableau `t` d'entiers, écrire un programme qui demande deux entiers `a` et `b` à l'utilisateur et annonce si `a` et `b` sont présents dans cet ordre (mais pas forcément consécutivement) dans `t`.

**Rendre le code du programme et les tests d'exécution**

## Exercice 8 (\*\*) : Exercice facultatif

Écrire un programme qui entrelace deux tableaux d'entiers dans un troisième tableau. Si les deux tableaux ne font pas la même longueur, les valeurs restantes du tableaux le plus grand sont ajoutées, dans l'ordre, au tableau de résultat. Par exemple, si `t1={1,5,3,6,7,8}` et `t2={1,2,4}` alors `tRes={1,1,5,2,3,4,6,7,8}`.

**Rendre le code du programme et les tests d'exécution**