

NADÉ  
Alexandre  
D2

**Ressource R1.01 TP 3**  
**Tableau**  
23/09/2022

Université de Bretagne Sud  
IUT de Vannes  
BUT Info 1D

## Exercice 1

### Conception méthodique

#### *Principe de l'algorithme*

Construire un programme qui saisit au maximum 10 entiers dans un tableau. La saisie peut être interrompue par l'entrée de la valeur -1. Le nombre de valeurs saisies doit être comptabilisé dans une variable 'nbVal'.

#### *Corps de boucle*

```
Value = SimpleInput.getInt(____) ;  
If (value ≠ 1) {  
    Tab[i] = value ;  
    i++ ;  
}
```

#### *Condition de sortie*

Value == 1 ou i >= tab.length

#### *Condition de continuation*

Value ≠ 1 && i < tab.length

#### *Initialisation*

```
Tab[] = new Int(10)  
Int i = 0 ; Int val = 0 ;
```

#### *Terminaison*

```
Int nbVal = i ;  
System.out.println(nbVal);
```

## Code

```
1. class ExOne {
2.     void principal() {
3.         int[] tab = new int[10];
4.         int i = 0;
5.         int value = 0;
6.
7.         while (value != -1 && i < tab.length) {
8.             value = SimpleInput.getInt("Donner une valeur");
9.             if (value != -1) {
10.                 tab[i] = value;
11.                 i = i + 1;
12.             }
13.         }
14.         int nbVal = i;
15.         System.out.println("Il y a " + nbVal + " :");
16.
17.     }
18. }
19.
```

## Réponse

Donner une valeur4  
Donner une valeur7  
Donner une valeur2  
Donner une valeur1  
Donner une valeur-1  
Il y a 4 valeurs

Donner une valeur1  
Donner une valeur2  
Donner une valeur3  
Donner une valeur4  
Donner une valeur5  
Donner une valeur6  
Donner une valeur7  
Donner une valeur8  
Donner une valeur9  
Donner une valeur10  
Il y a 10 valeurs

## Exercice 1 (2)

### Conception méthodique

#### Principe de l'algorithme

Modifier le programme précédent pour que le tableau soit affiché après avoir été saisi. L'affichage ne doit donc pas se faire au fur et à mesure de la saisie.

#### Corps de boucle

```
System.out.println(tab[j]);
```

```
j = j + 1;
```

#### Condition de sortie

```
j >= nbVal
```

#### Condition de continuation

```
j < nbVal
```

#### Initialisation

```
int j = 0 ;
```

### Code

```
1. class ExOne {
2.     void principal() {
3.         int[] tab = new int[10];
4.         int i = 0;
5.         int value = 0;
6.
7.         while (value != -1 && i < tab.length) {
8.             value = SimpleInput.getInt("Donner une valeur");
9.             if (value != -1) {
10.                tab[i] = value;
11.                i = i + 1;
12.            }
13.        }
14.        int nbVal = i;
15.        System.out.println("Il y a " + nbVal + " valeurs :");
16.
17.        int j = 0;
18.        while (j < nbVal) {
19.            System.out.println(tab[j]);
20.            j = j + 1;
21.        }
22.
23.    }
24. }
25.
```

### Réponse

```
Donner une valeur1
Donner une valeur2
Donner une valeur3
Donner une valeur4
Donner une valeur-1
Il y a 4 valeurs :
1
2
3
4
```

## Exercice 1 (3)

### Conception méthodique

#### Principe de l'algorithme

Modifier le programme pour qu'il demande une valeur et cherche si elle est présente dans le tableau déjà saisi.

#### Corps de boucle

```
if (tab[k] == valueCheck) {  
    nbValIn = nbValIn + 1;  
}
```

```
k = k + 1;
```

#### Condition de sortie

```
k >= nbVal
```

#### Condition de continuation

```
k < nbVal
```

#### Initialisation

```
int k = 0;
```

```
int nbValIn = 0;
```

### Code

```
1. class ExOne {  
2.     void principal() {  
3.         int[] tab = new int[10];  
4.         int i = 0;  
5.         int value = 0;  
6.         int j = 0;  
7.  
8.         while (value != -1 && i < tab.length) {  
9.             value = SimpleInput.getInt("Donner une valeur");  
10.  
11.             if (value != -1){  
12.                 tab[i] = value;  
13.                 i = i + 1;  
14.             }  
15.         }  
16.  
17.         int nbVal = i;  
18.         System.out.println("Il y a " + nbVal + " valeurs :");  
19.  
20.  
21.         while (j < nbVal) {  
22.             System.out.println(tab[j]);  
23.             j = j + 1;  
24.         }  
25.  
26.         int valueCheck = SimpleInput.getInt("Nombre est-il dans le tableau ?");  
27.  
28.         int k = 0;  
29.         int nbValIn = 0;  
30.         while (k < nbVal) {  
31.             if (tab[k] == valueCheck) {  
32.                 nbValIn = nbValIn + 1;  
33.             }  
34.             k = k + 1;  
35.         }  
36.  
37.         System.out.println(valueCheck + " se trouve " + nbValIn + " fois dans le tableau");  
38.     }  
39. }
```

## Réponse

```
Donner une valeur1
Donner une valeur2
Donner une valeur2
Donner une valeur3
Donner une valeur-1
Il y a 4 valeurs :
1
2
2
3
Nombre est-il dans le tableau ?3
3 se trouve 1 fois dans le tableau
```

```
Donner une valeur1
Donner une valeur2
Donner une valeur2
Donner une valeur3
Donner une valeur-1
Il y a 4 valeurs :
1
2
2
3
Nombre est-il dans le tableau ?2
2 se trouve 2 dans le tableau
```

## Exercice 2

### Conception méthodique

#### Principe de l'algorithme

Ajouter, au programme de saisie d'entiers dans un tableau, des lignes de code permettant de vérifier que les valeurs saisies sont triées dans l'ordre croissant.

#### Corps de boucle

```
if (tab[k] > tab[k + 1]) {  
    ascending = false;  
}
```

```
k = k + 1;
```

#### Condition de sortie

```
k >= (nbVal - 1)
```

#### Condition de continuation

```
k < (nbVal - 1)
```

#### Initialisation

```
int k = 0;  
ascending = true;
```

#### Code

```
1. class ExoTwo {  
2.     void principal() {  
3.         int[] tab = new int[10];  
4.         int i = 0;  
5.         int value = 0;  
6.         int j = 0;  
7.         boolean ascending = false;  
8.  
9.         while (value != -1 && i < tab.length) {  
10.            value = SimpleInput.getInt("Donner une valeur");  
11.  
12.            if (value != -1) {  
13.                tab[i] = value;  
14.                i = i + 1;  
15.            }  
16.        }  
17.        int nbVal = i;  
18.        System.out.println("Il y a " + nbVal + " valeurs :");  
19.  
20.        while (j < nbVal) {  
21.            System.out.println(tab[j]);  
22.            j = j + 1;  
23.        }  
24.        int k = 0;  
25.        ascending = true;  
26.  
27.        while (k < (nbVal - 1)) {  
28.            if (tab[k] > tab[k + 1]) {  
29.                ascending = false;  
30.            }  
31.            k = k + 1;  
32.        }  
33.        if (ascending) {  
34.            System.out.println("Croissant");  
35.        } else {  
36.            System.out.println("Décroissant");  
37.        }  
38.    }  
39. }
```

## Réponse

```
Donner une valeur1
Donner une valeur2
Donner une valeur3
Donner une valeur-1
Il y a 3 valeurs :
1
2
3
Croissant
```

```
Donner une valeur1
Donner une valeur9
Donner une valeur2
Donner une valeur3
Donner une valeur-1
Il y a 4 valeurs :
1
9
2
3
Décroissant
```

NADÉ  
Alexandre  
D2

**Ressource R1.01 TP 2**  
**Boucles**  
23/09/2022

Université de Bretagne Sud  
IUT de Vannes  
BUT Info 1D

## Exercice 1

### Conception méthodique

#### *Principe de l'algorithme*

Construire un programme qui saisit au maximum 10 entiers dans un tableau. La saisie peut être interrompue par l'entrée de la valeur -1. Le nombre de valeurs saisies doit être comptabilisé dans une variable 'nbVal'.

#### *Corps de boucle*

```
Value = SimpleInput.getInt(____) ;  
If (value ≠ 1) {  
    Tab[i] = value ;  
    i++ ;  
}
```

#### *Condition de sortie*

Value == 1 ou i >= tab.length

#### *Condition de continuation*

Value ≠ 1 && i < tab.length

#### *Initialisation*

```
Tab[] = new Int(10)  
Int i = 0 ; Int val = 0 ;
```

#### *Terminaison*

```
Int nbVal = i ;  
System.out.println(nbVal);
```



## Code

```
20. class ExOne {
21.     void principal() {
22.         int[] tab = new int[10];
23.         int i = 0;
24.         int value = 0;
25.
26.         while (value != -1 && i < tab.length) {
27.             value = SimpleInput.getInt("Donner une valeur");
28.             if (value != -1) {
29.                 tab[i] = value;
30.                 i = i + 1;
31.             }
32.         }
33.         int nbVal = i;
34.         System.out.println("Il y a " + nbVal + " :");
35.
36.     }
37. }
38.
```

## Réponse

Donner une valeur4  
Donner une valeur7  
Donner une valeur2  
Donner une valeur1  
Donner une valeur-1  
Il y a 4 valeurs

Donner une valeur1  
Donner une valeur2  
Donner une valeur3  
Donner une valeur4  
Donner une valeur5  
Donner une valeur6  
Donner une valeur7  
Donner une valeur8  
Donner une valeur9  
Donner une valeur10  
Il y a 10 valeurs

## Exercice 1 (2)

### Conception méthodique

#### Principe de l'algorithme

Modifier le programme précédent pour que le tableau soit affiché après avoir été saisi. L'affichage ne doit donc pas se faire au fur et à mesure de la saisie.

#### Corps de boucle

System.out.println(tab[j]);

j = j + 1;

#### Condition de sortie

j >= nbVal

#### Condition de continuation

j < nbVal

#### Initialisation

int j = 0 ;

### Code

```
26. class ExOne {
27.     void principal() {
28.         int[] tab = new int[10];
29.         int i = 0;
30.         int value = 0;
31.
32.         while (value != -1 && i < tab.length) {
33.             value = SimpleInput.getInt("Donner une valeur");
34.             if (value != -1) {
35.                 tab[i] = value;
36.                 i = i + 1;
37.             }
38.         }
39.         int nbVal = i;
40.         System.out.println("Il y a " + nbVal + " valeurs :");
41.
42.         int j = 0;
43.         while (j < nbVal) {
44.             System.out.println(tab[j]);
45.             j = j + 1;
46.         }
47.
48.     }
49. }
50.
```

### Réponse

```
Donner une valeur1
Donner une valeur2
Donner une valeur3
Donner une valeur4
Donner une valeur-1
Il y a 4 valeurs :
1
2
3
4
```

## Exercice 1 (3)

### Conception méthodique

#### Principe de l'algorithme

Modifier le programme pour qu'il demande une valeur et cherche si elle est présente dans le tableau déjà saisi.

#### Corps de boucle

```
if (tab[k] == valueCheck) {  
    nbValIn = nbValIn + 1;  
}
```

```
k = k + 1;
```

#### Condition de sortie

```
k >= nbVal
```

#### Condition de continuation

```
k < nbVal
```

#### Initialisation

```
int k = 0;
```

```
int nbValIn = 0;
```

### Code

```
40. class ExOne {  
41.     void principal() {  
42.         int[] tab = new int[10];  
43.         int i = 0;  
44.         int value = 0;  
45.         int j = 0;  
46.  
47.         while (value != -1 && i < tab.length) {  
48.             value = SimpleInput.getInt("Donner une valeur");  
49.  
50.             if (value != -1){  
51.                 tab[i] = value;  
52.                 i = i + 1;  
53.             }  
54.         }  
55.  
56.         int nbVal = i;  
57.         System.out.println("Il y a " + nbVal + " valeurs :");  
58.  
59.  
60.         while (j < nbVal) {  
61.             System.out.println(tab[j]);  
62.             j = j + 1;  
63.         }  
64.  
65.         int valueCheck = SimpleInput.getInt("Nombre est-il dans le tableau ?");  
66.  
67.         int k = 0;  
68.         int nbValIn = 0;  
69.         while (k < nbVal) {  
70.             if (tab[k] == valueCheck) {  
71.                 nbValIn = nbValIn + 1;  
72.             }  
73.             k = k + 1;  
74.         }  
75.  
76.         System.out.println(valueCheck + " se trouve " + nbValIn + " fois dans le tableau");  
77.     }  
78. }
```

## Réponse

```
Donner une valeur1
Donner une valeur2
Donner une valeur2
Donner une valeur3
Donner une valeur-1
Il y a 4 valeurs :
1
2
2
3
Nombre est-il dans le tableau ?3
3 se trouve 1 fois dans le tableau
```

```
Donner une valeur1
Donner une valeur2
Donner une valeur2
Donner une valeur3
Donner une valeur-1
Il y a 4 valeurs :
1
2
2
3
Nombre est-il dans le tableau ?2
2 se trouve 2 dans le tableau
```

## Exercice 2

### Conception méthodique

#### Principe de l'algorithme

Ajouter, au programme de saisie d'entiers dans un tableau, des lignes de code permettant de vérifier que les valeurs saisies sont triées dans l'ordre croissant.

#### Corps de boucle

```
if (tab[k] > tab[k + 1]) {  
    ascending = false;  
}
```

```
k = k + 1;
```

#### Condition de sortie

```
k >= (nbVal - 1)
```

#### Condition de continuation

```
k < (nbVal - 1)
```

#### Initialisation

```
int k = 0;  
ascending = true;
```

#### Code

```
40. class ExoTwo {  
41.     void principal() {  
42.         int[] tab = new int[10];  
43.         int i = 0;  
44.         int value = 0;  
45.         int j = 0;  
46.         boolean ascending = false;  
47.  
48.         while (value != -1 && i < tab.length) {  
49.             value = SimpleInput.getInt("Donner une valeur");  
50.  
51.             if (value != -1) {  
52.                 tab[i] = value;  
53.                 i = i + 1;  
54.             }  
55.         }  
56.         int nbVal = i;  
57.         System.out.println("Il y a " + nbVal + " valeurs :");  
58.  
59.         while (j < nbVal) {  
60.             System.out.println(tab[j]);  
61.             j = j + 1;  
62.         }  
63.         int k = 0;  
64.         ascending = true;  
65.  
66.         while (k < (nbVal - 1)) {  
67.             if (tab[k] > tab[k + 1]) {  
68.                 ascending = false;  
69.             }  
70.             k = k + 1;  
71.         }  
72.         if (ascending) {  
73.             System.out.println("Croissant");  
74.         } else {  
75.             System.out.println("Décroissant");  
76.         }  
77.     }  
78. }
```

## Réponse

```
Donner une valeur1
Donner une valeur2
Donner une valeur3
Donner une valeur-1
Il y a 3 valeurs :
1
2
3
Croissant
```

```
Donner une valeur1
Donner une valeur9
Donner une valeur2
Donner une valeur3
Donner une valeur-1
Il y a 4 valeurs :
1
9
2
3
Décroissant
```

## Exercice 3

### Conception méthodique

#### Principe de l'algorithme

Écrire un programme qui teste si la chaîne saisie est miroir (palindrome). Une chaîne est dite "miroir" si elle se lit de la même manière dans les deux sens. Par exemple "radar" est miroir ! Pour information : une chaîne Java, String, peut être vue comme un tableau de caractères. La variable mot est de type String

#### Corps de boucle

```
currentLetter = mot.charAt(i);  
motInverse = String.valueOf(currentLetter) + motInverse;  
i = i + 1;
```

#### Condition de sortie

$i \geq j$

#### Condition de continuation

$i < j$

#### Initialisation

```
int j = mot.length();  
int i = 0;
```

#### Code

```
1. class ExoThree {  
2.     void principal() {  
3.  
4.         String mot;  
5.  
6.         mot = SimpleInput.getString("Mot ? ");  
7.  
8.         char currentLetter;  
9.         String motInverse = "";  
10.  
11.        boolean palindrome = false;  
12.  
13.        int j = mot.length();  
14.        int i = 0;  
15.        while (i < j) {  
16.            currentLetter = mot.charAt(i);  
17.            motInverse = String.valueOf(currentLetter) + motInverse;  
18.            i = i + 1;  
19.        }  
20.  
21.        if (mot.equals(motInverse)) {  
22.            palindrome = true;  
23.        }  
24.  
25.        if (palindrome) {  
26.            System.out.println("Le mot est un palindrome");  
27.        } else {  
28.            System.out.println("Le mot n'est pas un palindrome");  
29.        }  
30.    }  
31. }  
32.
```

## Réponse

Mot ? radar  
Le mot est un palindrome

Mot ? kayak  
Le mot est un palindrome

Mot ? mot  
Le mot n'est pas un palindrome



## Exercice 4

### Conception méthodique

#### Principe de l'algorithme

Ajouter, au programme de saisie d'entiers dans un tableau, des lignes de code permettant de vérifier que les valeurs saisies sont triées dans l'ordre croissant.

#### Corps de boucle

```
if (tab[k] == valueCheck) {  
    nbValIn = nbValIn + 1;  
}  
k = k + 1;
```

```
if (nbValIn >= 2) {  
    twoVal = true;  
}
```

#### Condition de sortie

k >= nbVal

#### Condition de continuation

k < (nbVal - 1)

#### Initialisation

```
int k = 0;  
int nbValIn = 0;  
boolean twoVal = false;
```

### Code

```
1. class ExoFour {  
2.     void principal() {  
3.         int[] tab = new int[10];  
4.         int i = 0;  
5.         int value = 0;  
6.         int j = 0;  
7.  
8.         while (value != -1 && i < tab.length) {  
9.             value = SimpleInput.getInt("Donner une valeur");  
10.  
11.             if (value != -1) {  
12.                 tab[i] = value;  
13.                 i = i + 1;  
14.             }  
15.         }  
16.  
17.  
18.         int nbVal = i;  
19.         System.out.println("Il y a " + nbVal + " valeurs :");  
20.  
21.  
22.         while (j < nbVal) {  
23.             System.out.println(tab[j]);  
24.             j = j + 1;  
25.         }  
26.  
27.  
28.         int valueCheck = SimpleInput.getInt("Nombre est t-il dans le tableau ?");  
29.  
30.         int k = 0;  
31.         int nbValIn = 0;  
32.         boolean twoVal = false;  
33.         while (k < nbVal) {
```

```

34.         if (tab[k] == valueCheck) {
35.             nbValIn = nbValIn + 1;
36.         }
37.         k = k + 1;
38.
39.         if (nbValIn >= 2) {
40.             twoVal = true;
41.         }
42.     }
43.
44.     if (twoVal){
45.         System.out.println(valueCheck + " est présent en au moins deux exemplaires
dans le tableau d'entiers");
46.     }
47. }
48. }
49.

```

## Réponse

Donner une valeur1  
 Donner une valeur2  
 Donner une valeur3  
 Donner une valeur-1  
 Il y a 3 valeurs :  
 1  
 2  
 3  
 Croissant

Donner une valeur1  
 Donner une valeur9  
 Donner une valeur2  
 Donner une valeur3  
 Donner une valeur-1  
 Il y a 4 valeurs :  
 1  
 9  
 2  
 3  
 Décroissant