

NADÉ **Ressource R1.01 TP 3** Université de Bretagne Sud

Alexandre **Tableau** IUT de Vannes

D2 23/09/2022 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

#### Iniatialisation

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;
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 + " :");
17. }
18. }

**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

#### Iniatialisation

int j = 0 ;

**Code**

1. class ExOne {
2. void principal() {
3. int[] tab = new int[10];
4. int i = 0;
5. int value = 0;
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 :");
17. int j = 0;
18. while (j < nbVal) {
19. System.out.println(tab[j]);
20. j = j + 1;
21. }
23. }
24. }

**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

#### Iniatialisation

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;
8. while (value != -1 && i<tab.length) {
9. value = SimpleInput.getInt("Donner une valeur");
11. if (value != -1){
12. tab[i] = value;
13. i = i + 1;
14. }
15. }
17. int nbVal = i;
18. System.out.println("Il y a " + nbVal + " valeurs :");

21. while (j < nbVal) {
22. System.out.println(tab[j]);
23. j = j + 1;
24. }
26. int valueCheck = SimpleInput.getInt("Nombre est-il dans le tableau ?");
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. }
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 permettantde 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)

#### Iniatialisation

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;
9. while (value != -1 && i<tab.length) {
10. value = SimpleInput.getInt("Donner une valeur");
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 :");
20. while (j < nbVal) {
21. System.out.println(tab[j]);
22. j = j + 1;
23. }
24. int k = 0;
25. ascending = true;
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 :

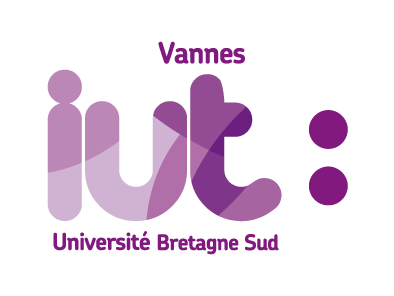
1

9

2

3

Décroissant



NADÉ **Ressource R1.01 TP 2** Université de Bretagne Sud

Alexandre **Boucles** IUT de Vannes

D2 23/09/2022 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

#### Iniatialisation

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;
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 + " :");
17. }
18. }

**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

#### Iniatialisation

int j = 0 ;

**Code**

1. class ExOne {
2. void principal() {
3. int[] tab = new int[10];
4. int i = 0;
5. int value = 0;
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 :");
17. int j = 0;
18. while (j < nbVal) {
19. System.out.println(tab[j]);
20. j = j + 1;
21. }
23. }
24. }

**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

#### Iniatialisation

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;
8. while (value != -1 && i<tab.length) {
9. value = SimpleInput.getInt("Donner une valeur");
11. if (value != -1){
12. tab[i] = value;
13. i = i + 1;
14. }
15. }
17. int nbVal = i;
18. System.out.println("Il y a " + nbVal + " valeurs :");

21. while (j < nbVal) {
22. System.out.println(tab[j]);
23. j = j + 1;
24. }
26. int valueCheck = SimpleInput.getInt("Nombre est-il dans le tableau ?");
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. }
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 permettantde 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)

#### Iniatialisation

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;
9. while (value != -1 && i<tab.length) {
10. value = SimpleInput.getInt("Donner une valeur");
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 :");
20. while (j < nbVal) {
21. System.out.println(tab[j]);
22. j = j + 1;
23. }
24. int k = 0;
25. ascending = true;
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

# Exercice 3

### Conception méthodique

#### Principe de l’algorithme

Écrire un programme qui teste si la chaîne saisie est miroir(palindrome). Une chaine 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 *>=* *j*

#### Condition de continuation

#### i < j

#### Iniatialisation

int j = mot.length();

int i = 0;

**Code**

1. class ExoThree {
2. void principal() {
4. String mot;
6. mot = SimpleInput.getString("Mot ? ");
8. char currentLetter;
9. String motInverse = "";
11. boolean palindrome = false;
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. }
21. if (mot.equals(motInverse)) {
22. palindrome = true;
23. }
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. }

**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 permettantde 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)

#### Iniatialisation

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;
8. while (value != -1 && i<tab.length) {
9. value = SimpleInput.getInt("Donner une valeur");
11. if (value != -1){
12. tab[i] = value;
13. i = i + 1;
14. }
15. }

18. int nbVal = i;
19. System.out.println("Il y a " + nbVal + " valeurs :");

22. while (j < nbVal) {
23. System.out.println(tab[j]);
24. j = j + 1;
25. }

28. int valueCheck = SimpleInput.getInt("Nombre est t-il dans le tableau ?");
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;
39. if (nbValIn >= 2) {
40. twoVal = true;
41. }
42. }
44. if (twoVal){
45. System.out.println(valueCheck + " est présent en au moins deux exemplaires dans le tableau d’entiers");
46. }
47. }
48. }

**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