

# Algorithmique

## 5. Les tableaux

1

### Plan du cours

---

- I. Introduction
- II. Variables
- III. Structures alternatives
- IV. Structures itératives
- V. Tableaux**
- VI. Fonctions et procédures
- VII. Fichiers

▶ 2

Algorithmique ESI 2023-2024

2

## Sommaire

---

- I. Tableaux statiques
- II. Tableaux dynamiques
- III. Tri dans un tableau
- IV. Recherche dans un tableau

---

▶ 3

Algorithmique ESI 2023-2024

3

## Problématique

---

- ▶ Les boucles permettent à l'utilisateur de saisir **n** nombres → on déclare une seule variable mais on écrase sa valeur à chaque nouvelle saisie
- ▶ Si on veut garder les **n** nombres pour les utiliser par la suite → on déclare **n** variables différentes
  - ▶ Quand **n** est petit → Lourd mais gérable
  - ▶ Quand **n** est grand ?

---

▶ 4

Algorithmique ESI 2023-2024

4

## Problématique

- ▶ Les boucles permettent à l'utilisateur de saisir **n** nombres → on déclare une seule variable mais on écrase sa valeur à chaque nouvelle saisie
- ▶ Si on veut garder les **n** nombres pour les utiliser par la suite → on déclare **n** variables différentes
- ▶ Idéalement, tout stocker dans une seule variable → **tableau**

▶ 5

Algorithmique

ESI

2023-2024

5

## Tableaux statiques

6

## Définition

- ▶ Tableau statique
  - ▶ Séquence de données de **même type**, chacune référencée par un nombre appelé **indice** (ou index)
  - ▶ Désigné par
    - ▶ Son nom
    - ▶ Le type de ses éléments
    - ▶ Sa taille (i.e. le nombre de ses éléments)

▶ 7

Algorithmique

ESI

2023-2024

7

## Déclaration de tableaux statiques

- ▶ Syntaxe
 

**VAR** nomTableau[N] :Type

  - ▶ N : taille du tableau
  - ▶ Premier indice : 0
  - ▶ Dernier indice :  $N - 1$

▶ 8

Algorithmique

ESI

2023-2024

8

## Déclaration de tableaux statiques

### ► Syntaxe

**VAR** nomTableau[N] :Type

### ► Exemple :

**VAR** nombres[5] : entier

**VAR** lettres[26] : caractère

**VAR** mots[100] : chaîne

**VAR** absents[107] : booléen

► 9

Algorithmique ESI 2023-2024

9

## Utilisation de tableaux statiques

- Pour désigner un élément du tableau, on utilise son nom avec, entre crochets, l'indice de l'élément concerné

### ► Exemple :

nombres[0]

nombres[1]

nombres[2]

nombres[3]

nombres[4]

► 10

Algorithmique ESI 2023-2024

10

## Utilisation de tableaux statiques

- Pour affecter une valeur à une case du tableau, on utilise son nom avec, entre parenthèses, l'indice concerné puis le signe d'affectation et la valeur

- Exemple :

nombres[0] ← 5

nombres[1] ← 60

nombres[2] ← 10

nombres[3] ← 0

nombres[4] ← 90

► I1

Algorithmique

ESI

2023-2024

11

## Utilisation de tableaux statiques

nombres[0] ← 5

nombres[1] ← 60

nombres[2] ← 10

nombres[3] ← 0

nombres[4] ← 90

0	1	2	3	4
5	60	10	0	90

► I2

Algorithmique

ESI

2023-2024

12

## Utilisation de tableaux statiques

- ▶ L'indice peut être un nombre, une variable ou une expression calculée
- ▶ Exemple

```
nombres[0]
nombres[i]
nombres[2*i+5]
```

▶ I3

Algorithmique

ESI

2023-2024

13

## Utilisation de tableaux statiques

- ▶ Exemple : demander à l'utilisateur de saisir 10 nombres réels

```
ALGORITHME tableau_réels
VAR nombres[10] : réel
VAR i : entier
DEBUT
  POUR i ← 0 à 9
    Afficher(" Saisir le nombre numéro ", i+1)
    Lire(nombres[i])
  i SUIVANT
FIN
```

▶ I4

Algorithmique

ESI

2023-2024

14

## Exercice

- Que fait cet algorithme ?

```

ALGORITHME exemple_tableau
VAR tableau[10], i : entier
DEBUT
  POUR i  $\leftarrow$  0 à 9
    tableau[i]  $\leftarrow$  i
  i SUIVANT
  POUR i  $\leftarrow$  0 à 9
    Afficher(tableau[i])
  i SUIVANT
FIN

```

► I5

Algorithmique

ESI

2023-2024

15

## Exercice

- Que fait cet algorithme ?

```

ALGORITHME exemple_tableau
VAR tableau[10], i : entier
DEBUT
  tableau[0]  $\leftarrow$  0
  POUR i  $\leftarrow$  1 à 9
    tableau[i]  $\leftarrow$  tableau[i - 1] + 1
  i SUIVANT
  POUR i  $\leftarrow$  0 à 9
    Afficher(tableau[i])
  i SUIVANT
FIN

```

► I6

Algorithmique

ESI

2023-2024

16



## Exercice

- ▶ Écrire un algorithme qui demande à l'utilisateur de saisir 20 nombres et puis qui les affiche

▶ I7

Algorithmique

ESI

2023-2024

17

## Exercice

**ALGORITHME** *moyenne*

**VAR** nombres[20] : **réel**

**VAR** i : **entier**

**DEBUT**

**POUR** i  $\leftarrow$  0 à 19

    Afficher("Saisir le nombre numéro ", i+1)

    Lire(nombres[i])

  i **SUIVANT**

**POUR** i  $\leftarrow$  0 à 19

    Afficher("Le ", i+1, "ème nombre est : ", nombres[i])

  i **SUIVANT**

**FIN**

▶ I8

Algorithmique

ESI

2023-2024

18

## Exercice

- ▶ Écrire un algorithme qui demande à l'utilisateur de saisir 20 nombres et puis qui les affiche à partir du dernier saisi

▶ 19

Algorithmique

ESI

2023-2024

19

## Exercice

```

ALGORITHME moyenne
VAR nombres[20] : réel
VAR i : entier
DEBUT
  POUR i ← 0 à 19
    Afficher("Saisir le nombre numéro ", i+1)
    Lire(nombres[i])
  i SUIVANT
  POUR i ← 19 à 0 pas – 1
    Afficher("Le ", i+1, "ème nombre est : ", nombres[i])
  i SUIVANT
FIN
  
```

▶ 20

Algorithmique

ESI

2023-2024

20

## Exercice

- ▶ Écrire un algorithme qui demande à l'utilisateur de saisir 20 notes et lui affiche leur moyenne

▶ 21

Algorithmique

ESI

2023-2024

21

## Exercice

```

ALGORITHME moyenne
VAR notes[20], somme : réel
VAR i : entier
DEBUT
    somme ← 0
    POUR i ← 0 à 19
        Afficher(" Saisir la note numéro ", i)
        Lire(notes[i])
        somme ← somme + notes[i]
    SUIVANT i
    Afficher("La moyenne est : ", somme/20)
FIN
  
```

▶ 22

Algorithmique

ESI

2023-2024

22

## Exercice

- ▶ Écrire un algorithme qui demande à l'utilisateur de saisir 20 notes et lui affiche leur moyenne, la note minimale et la note maximale

▶ 23

Algorithmique

ESI

2023-2024

23

```

ALGORITHME moyenne_min_max
VAR notes[20], somme, min, max : réel
VAR i : entier
DEBUT
  Afficher("Saisir la 1ère note")
  Lire(notes[0])
  somme, max, min ← notes[0]
  POUR i ← 1 à 19
    Afficher(" Saisir la note numéro ", i+1)
    Lire(notes[i])
    somme ← somme + notes[i]
    SI notes[i] < min ALORS
      min ← notes[i]
    SINONSI notes[i] > max ALORS
      max ← notes[i]
    FINSI
  i SUIVANT
  Afficher("Moyenne = ", somme/20, "Min = ", min, "Max = ", max)
FIN

```

24

## Problématique

- ▶ Écrire un algorithme qui demande à l'utilisateur de saisir des notes et lui affiche leur moyenne, la note minimale et la note maximale
- ▶ Le nombre de notes à saisir n'est pas connu à l'avance

➔ **Tableau dynamique**

▶ 25

Algorithmique

ESI

2023-2024

25

## Tableaux dynamiques

26

## Définition

### ► Tableau dynamique

- Séquence de données de **même type**, chacune référencée par un nombre appelé **indice** (ou index), dont la **taille est variable** et peut changer lors de l'exécution

► 27

Algorithmique

ESI

2023-2024

27

## Déclaration de tableaux dynamiques

### ► Syntaxe

**VAR** nomTableau[ ] :Type

### ► Exemple :

**VAR** nombres[ ] : entier

**VAR** lettres[ ] : caractère

**VAR** mots[ ] : chaîne

**VAR** absents[ ] : booléen

► 28

Algorithmique

ESI

2023-2024

28

## Redimensionnement

- ▶ L'opération de redimensionnement permet de fixer la taille du tableau dynamique
- ▶ Elle intervient dans le corps de l'algorithme
- ▶ Elle est effectuée par l'instruction **Redim**

▶ 29

Algorithmique

ESI

2023-2024

29

## Redimensionnement

```

ALGORITHME tableau_dynamique
VAR notes[ ] : réel
VAR n,i : entier
DEBUT
    Afficher("Quelle est le nombre de notes à saisir ?")
    Lire(n)
    Redim notes[n]
    POUR i ← 0 à n-1
        Afficher(" Saisir la note numéro ", i + 1)
        Lire(notes[i])
    i SUIVANT
FIN
  
```

▶ 30

Algorithmique

ESI

2023-2024

30

## Redimensionnement

- ▶ L'opération de redimensionnement permet de fixer la taille du tableau dynamique
- ▶ Elle intervient dans le corps de l'algorithme
- ▶ Elle est effectuée par l'instruction **Redim**
  - ▶ À tout moment, on peut ajouter / supprimer des cases

▶ 31

Algorithmique

ESI

2023-2024

31

## Exercice

- ▶ Écrire un algorithme qui demande à l'utilisateur de saisir des notes et lui affiche leur moyenne, la note minimale et la note maximale
  - ▶ Demander à l'utilisateur combien de notes il veut saisir

▶ 32

Algorithmique

ESI

2023-2024

32



```

ALGORITHME moyenne
VAR notes[ ], somme, min, max : réel
VAR n, i : entier
DEBUT
    somme ← 0
    Afficher("Combien de notes voulez-vous saisir ?")
    Lire(n)
    Redim notes[n]
    Afficher("Saisir la 1ère note")
    Lire(notes[0])
    somme, max, min ← notes[0]
    POUR i ← 1 à n-1
        Afficher(" Saisir la note numéro ", i)
        Lire(notes[i])
        somme ← somme + notes[i]
        SI notes[i] < min ALORS
            min ← notes[i]
        SINONSI notes[i] > max ALORS
            max ← notes[i]
    FINSI
    i SUIVANT
    Afficher("Moyenne = ", somme/n, "Min = ", min, "Max = ", max)
FIN

```

33

## Exercice

- ▶ Écrire un algorithme qui demande à l'utilisateur de saisir des nombres réels puis qui lui affiche le nombre des positifs, le nombre des négatifs, le nombre minimal et le nombre maximal
- ▶ Demander à l'utilisateur combien de nombres il veut saisir

34

```

ALGORITHME exercice_tableau_dynamique
VAR nombres[ ], min, max : réel
VAR n, i, nbPositifs : entier
DEBUT
  nbPositifs ← 0
  Afficher("Combien de nombre voulez-vous saisir ?")
  Lire(n)
  Redim nombres[n]
  POUR i ← 0 à n-1
    Afficher("Saisir le nombre numéro ", i)
    Lire(nombres[i])
    SI i = 0 ALORS
      min ← nombres[i]
      max ← nombres[i]
    SINONSI nombres[i] < min ALORS
      min ← nombres[i]
    SINONSI nombres[i] > max ALORS
      max ← nombres[i]
    FINSI
    SI nombres[i] >= 0 ALORS
      nbPositifs ← nbPositifs + 1
    FINSI
  i SUIVANT
  Afficher("Min = ", min, "Max = ", max, "Nb + = ", nbPositifs, "Nb - = ", n - nbPositifs)
FIN

```

35

## Exercice

- ▶ Écrire un algorithme qui demande à l'utilisateur de remplir deux tableaux (de même taille) puis qui affiche le tableau contenant le produit des éléments des deux

- ▶ Demander à l'utilisateur la taille des tableaux

Tableau 1	5	2	10	15	1
✕ Tableau 2	5	60	10	0	90
Tableau 3	25	120	100	0	90

▶ 36

Algorithmique ESI 2023-2024

36

```

ALGORITHME exercice_tableau_dynamique
VAR tab1[ ], tab2[ ], tab3[ ] : réel
VAR n, i : entier
DEBUT
  Afficher("Combien de nombre voulez-vous saisir ?")
  Lire(n)
  Redim tab1[n]
  Redim tab2[n]
  Redim tab3[n]
  Afficher("Remplissage du 1er tableau")
  POUR i ← 0 à n-1
    Afficher("Saisir le nombre numéro ", i+1)
    Lire(tab1[i])
  i SUIVANT
  Afficher("Remplissage du 2e tableau")
  POUR i ← 0 à n-1
    Afficher("Saisir le nombre numéro ", i+1)
    Lire(tab2[i])
  i SUIVANT
  POUR i ← 0 à n-1
    tab3[i] ← tab1[i]*tab2[i]
    Afficher(tab1[i], "x", tab2[i], "=", tab3[i])
  i SUIVANT
FIN

```

37

## Problématique

- ▶ **Besoin** : pour une classe de 10 étudiants, nous voulons stocker, pour chaque étudiant, ses notes dans 2 matières
  - ▶ Un tableau de 10 éléments pour la note de la première matière; et un autre tableau pour la note de la deuxième
  - ▶ Matrice de 10 x 2 avec, dans chaque ligne i, les deux notes de l'étudiant numéro i → **tableau multidimensionnel**

▶ 38

Algorithmique

ESI

2023-2024

38

## Tableaux multidimensionnels

39

### Définition

- ▶ Tableau à 2 dimensions
  - ▶ Séquence de données de **même type**, chacune référencée par **deux indices**
  - ▶ Désigné par
    - ▶ Son nom
    - ▶ Le type de ses éléments
    - ▶ Sa taille  $n \times m$ 
      - Peut être assimilé à une matrice de  $n$  lignes et  $m$  colonnes

▶ 40

Algorithmique ESI 2023-2024

40

## Déclaration de tableaux à 2 dimensions

### ► Syntaxe

**VAR** nomTableau[N][M] :Type

### ► Exemple :

**VAR** nombres[5][5] : entier

**VAR** lettres[26][2] : caractère

**VAR** mots[1][10] : chaîne

**VAR** absents[107][30] : booléen

► 41

Algorithmique

ESI

2023-2024

41

## Exemple de tableaux à 2 dimensions

**ALGORITHME** *tableau\_2d*

**VAR** notes[20][2] : **réel**

**DEBUT**

**POUR** i ← 0 à 19

    Afficher("Saisir la 1<sup>ère</sup> note de l'étudiant N° ", i+ 1)

    Lire(notes[i][0])

    Afficher("Saisir la 2<sup>e</sup> note de l'étudiant N° ", i+ 1)

    Lire(notes[i][1])

  i **SUIVANT**

**FIN**

► 42

Algorithmique

ESI

2023-2024

42

## Exercice

- ▶ Que font les algorithmes suivants ?

▶ 43

Algorithmique

ESI

2023-2024

43

**ALGORITHME** *exemple\_tab*

**VAR** *tab*[5][10] : **entier**

**VAR** *i, j, k*: **entier**

**DEBUT**

$k \leftarrow 1$

**POUR**  $i \leftarrow 0 \text{ à } 4$

**POUR**  $j \leftarrow 0 \text{ à } 9$

$\text{tab}[i][j] \leftarrow k$

$k \leftarrow k + 1$

$j$  **SUIVANT**

$i$  **SUIVANT**

**FIN**

44

**ALGORITHME** *exemple\_tab*

**VAR** tab[5][10] : entier

**VAR** i, j : entier

**DEBUT**

**POUR** i  $\leftarrow$  0 à 4

**POUR** j  $\leftarrow$  0 à 9

            tab[i][j]  $\leftarrow$  i + j

        j **SUIVANT**

    i **SUIVANT**

**FIN**

45

## Exercice

- ▶ Écrire un algorithme qui génère un tableau de booléens de 5 x 10 où toutes les cases contiennent la valeur VRAI

46

## Exercice

```

ALGORITHME tab_bool
VAR tab[5][10] : booléen
DEBUT
  POUR i ← 0 à 4
    POUR j ← 0 à 9
      tab[i][j] ← VRAI
    j SUIVANT
  i SUIVANT
FIN

```

▶ 47

Algorithmique

ESI

2023-2024

47

## Exercice

- ▶ Écrire un algorithme qui demande à l'utilisateur de remplir une matrice de 10 x 10 et vérifie si elle est symétrique ou non

▶ 48

Algorithmique

ESI

2023-2024

48



## Exercice

- Écrire un algorithme qui demande à l'utilisateur de remplir une matrice de 10 x 10 et vérifie si elle est symétrique ou non

x	a	b	c
a	y	d	e
b	d	z	f
c	e	f	q

► 49

Algorithmique

ESI

2023-2024

49

## Exercice

```

ALGORITHME symétrie_matrice
VAR matrice[10][10] : réel
VAR i, j : entier
VAR sym : booléen
DEBUT
  sym ← VRAI
  POUR i ← 0 à 9
    POUR j ← 0 à 9
      Lire(matrice[i][j])
    SUIVANT j
  SUIVANT i

```

► 50

Algorithme

```

i ← 0
TANTQUE sym ET i < 10
  j ← 0
  TANTQUE sym ET j < 10
    SI j <> i et matrice[i][j] <> matrice[j][i] ALORS
      sym ← FALSE
    FINSI
    j ← j + 1
  FINTANTQUE
  i ← i + 1
FINTANTQUE
SI sym ALORS
  Afficher("La matrice saisie est symétrique")
SINON
  Afficher("La matrice saisie n'est pas symétrique")
FINSI
FIN

```

50

## Exercice

- ▶ Écrire un algorithme qui cherche, dans un tableau d'entiers de 10 x 10, le nombre minimal et le nombre maximal

▶ 51

Algorithmique

ESI

2023-2024

51

```

ALGORITHME tab_bool
VAR tab[10][10], min, max : réel
VAR i, j : entier
DEBUT
  //instructions de remplissage du tableau omises
  min, max ← tab[0][0]
  POUR i ← 0 à 9
    POUR j ← 0 à 9
      SI tab[i][j] < min ALORS
        min ← tab[i][j]
      SINONSI tab[i][j] > max ALORS
        max ← tab[i][j]
      FINSI
    j SUIVANT
  i SUIVANT
  Afficher("Max = ", max, " et Min = ", min)
FIN

```

▶ 52

Algorithmique

ESI

2023-2024

52

## Recherche dans un tableau

53

### Recherche séquentielle

- ▶ Vérifier si une valeur existe dans un tableau → utiliser un flag
  - ▶ Parcourir le tableau
    - ▶ Si élément courant = valeur → flag = VRAI
  - ▶ Vérifier le flag à la fin de l'exécution
    - ▶ Si flag = VRAI → l'élément existe

▶ 54

Algorithmique

ESI

2023-2024

54

## Recherche séquentielle

### ► Pseudo-code

- Recherche de la valeur *val* dans le tableau `tab[N]`

► 55

Algorithmique

ESI

2023-2024

55

## Recherche séquentielle

existe  $\leftarrow$  FAUX

**POUR**  $i \leftarrow 0$  à  $N-1$

**SI** `tab[i] = val` **ALORS**

        existe  $\leftarrow$  VRAI

**FINSI**

$i$  **SUIVANT**

**SI** existe **ALORS**

    Afficher("La valeur recherchée a été trouvée")

**SINON**

    Afficher("La valeur recherchée n'a pas été trouvée")

**FINSI**

► 56

Algorithmique

ESI

2023-2024

56

## Recherche séquentielle

- ▶ Pseudo-code
  - ▶ Recherche de la valeur *val* dans le tableau `tab[N]`
  - ▶ Amélioration possible : ne pas continuer les comparaisons si l'élément a été trouvé
  - ▶ Encore mieux : **recherche dichotomique**

▶ 57

Algorithmique

ESI

2023-2024

57

## Recherche dichotomique

- ▶ Principe
  - ▶ Comparer la valeur à trouver avec l'élément au milieu du tableau
    - ▶ Si inférieure, refaire la même comparaison avec la première moitié du tableau
    - ▶ Si supérieure, refaire la même comparaison avec la deuxième moitié du tableau
  - ▶ Nécessite que le tableau soit trié

▶ 58

Algorithmique

ESI

2023-2024

58

## Recherche dichotomique

### ► Pseudo-code

- Recherche de la valeur *val* dans le tableau `tab[N]`

► 59

Algorithmique

ESI

2023-2024

59

## Recherche dichotomique

```

existe ← FAUX
debut ← 0
fin ← N-1
TANTQUE existe = FAUX ET debut ≤ fin
    milieu ← (debut + fin) DIV 2
    SI tab[milieu] = val ALORS
        existe ← VRAI
    SINONSI tab[milieu] < val ALORS
        debut ← milieu + 1
    SINON
        fin ← milieu - 1
FINSI
FINTANTQUE

```

► 60

Algorithmique

ESI

2023-2024

60

## Tri dans un tableau

61

### Motivation

- ▶ Il est souvent nécessaire de stocker les valeurs d'un tableau selon un **ordre** déterminé (ascendant ou descendant)
- ▶ Plusieurs approches possibles
  - ▶ Tri par sélection
  - ▶ Tri à bulles
  - ▶ ...

▶ 62

Algorithmique

ESI

2023-2024

62

## Tri par sélection

- ▶ Approche intuitive
  - ▶ On place le plus petit élément à la position 0
  - ▶ Pui on place le plus petit des éléments restant à la position 1
  - ▶ Puis on place le plus petit des éléments restant à la position 2
  - ▶ Et ainsi de suite jusqu'au dernier élément

▶ 63

Algorithmique

ESI

2023-2024

63

## Tri par sélection

- ▶ Approche intuitive
  - ▶ On détermine le plus petit élément, noté min, à partir de l'indice  $i$  (à commencer par  $i = 0$ , le 1<sup>er</sup> élément)
  - ▶ Échanger min et l'élément de la case  $i$
  - ▶ Refaire la même chose avec  $i + 1$  jusqu'à avoir trié tous les éléments

▶ 64

Algorithmique

ESI

2023-2024

64



## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
---	---	----	----	---

► 65

Algorithmique

ESI

2023-2024

65

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
---	---	----	----	---

► 66

Algorithmique

ESI

2023-2024

66

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
---	---	----	----	---

► 67

Algorithmique

ESI

2023-2024

67

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
---	---	----	----	---

1	2	10	15	5
---	---	----	----	---

► 68

Algorithmique

ESI

2023-2024

68

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
1	2	10	15	5

► 69

Algorithmique

ESI

2023-2024

69

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
1	2	10	15	5
1	2	10	15	5

► 70

Algorithmique

ESI

2023-2024

70

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5

► 71

Algorithmique

ESI

2023-2024

71

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5

► 72

Algorithmique

ESI

2023-2024

72

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5
1	2	5	15	10

► 73

Algorithmique

ESI

2023-2024

73

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5
1	2	5	15	10
1	2	5	15	10

► 74

Algorithmique

ESI

2023-2024

74

## Tri par sélection

### ► Approche intuitive

5	2	10	15	1
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5
1	2	10	15	5
1	2	5	15	10
1	2	5	15	10
1	2	5	10	15

► 75

75

## Tri par sélection

### ► Pseudo-code

#### ► Exemple du tri par sélection du tableau $\text{tab}[N]$

► 76

Algorithmique ESI 2023-2024

76

```

POUR  $i \leftarrow 0$  à  $N - 2$ 
    indice_min  $\leftarrow i$  //prendre l'élément tab[i] comme plus petit élément provisoire
    //examiner tous les éléments suivants
    POUR  $j \leftarrow i + 1$  à  $N - 1$ 
        SI  $\text{tab}[j] < \text{tab}[\text{indice\_min}]$  ALORS
            indice_min  $\leftarrow j$  //récupérer l'indice de l'élément plus petit que tab[i]
        FINSI
    j SUIVANT
    //permuter tab[i] et l'élément le plus petit, qui est tab[indice_min]
     $x \leftarrow \text{tab}[i]$ 
     $\text{tab}[i] \leftarrow \text{tab}[\text{indice\_min}]$ 
     $\text{tab}[\text{indice\_min}] \leftarrow x$ 
i SUIVANT //passer à l'élément tab[i+1]

```

▶ 77

Algorithmique

ESI

2023-2024

77

## Tri par sélection

### ▶ Pseudo-code

- ▶ Variante possible : effectuer la permutation au fur et à mesure

▶ 78

Algorithmique

ESI

2023-2024

78

## Tri par sélection

```

POUR  $i \leftarrow 0$  à  $N - 2$ 
    //examiner tous les éléments suivants
    POUR  $j \leftarrow i + 1$  à  $N - 1$ 
        SI  $\text{tab}[j] < \text{tab}[i]$  ALORS //permuter à ce stade
             $x \leftarrow \text{tab}[i]$ 
             $\text{tab}[i] \leftarrow \text{tab}[j]$ 
             $\text{tab}[j] \leftarrow x$ 
        FINSI
     $j$  SUIVANT
i SUIVANT //passer à l'élément  $\text{tab}[i+1]$ 
  
```

▶ 79

Algorithmique

ESI

2023-2024

79

## Tri à bulles

- ▶ Approche
  - ▶ On initialise un flag à FAUX
  - ▶ On compare chaque élément à celui qui le suit
    - ▶ Si non triés, on les permute et on met le flag à VRAI

▶ 80

Algorithmique

ESI

2023-2024

80



## Tri à bulles

### ► Approche

#### ► À la sortie de la boucle

- Si flag = FAUX → le tableau est trié
- Si flag = VRAI → au moins une permutation a eu lieu → refaire une itération pour vérifier s'il reste d'autres permutations à faire

► 81

Algorithmique

ESI

2023-2024

81

## Tri à bulles

### ► Pseudo-code

#### ► Exemple du tri à bulles du tableau `tab[N]`

► 82

Algorithmique

ESI

2023-2024

82

# Tri à bulles

```

permutation ← VRAI
TANTQUE permutation
    permutation ← FAUX
    POUR i ← 0 à N - 2
        SI tab[i+1] < tab[i] ALORS
            x ← tab[i]
            tab[i] ← tab[i+1]
            tab[i+1] ← x
        permutation ← VRAI
    FINSI
SUIVANT i
FINTANTQUE

```

83

## Algorithmique

ESI

2023-2024

83



مدرسة علوم المعلومات  
+212 31 20 00 01 | 31 24 24 01  
ECOLE DES SCIENCES  
DE L'INFORMATION  
[www.esi.ac.ma](http://www.esi.ac.ma)

# Algorithmique

## 5. Les tableaux

84