

Introduction à l'informatique CM3

Antonio E. Porreca
aeporreca.org/introinfo

Décrire des algorithmes

- En **langage naturel** (par exemple, en français avec un accent italien)
- En **pseudocode** (semi-formel)
- En **langage de programmation** (formel)
 - Par exemple, en Python dans l'UE
 - Mise en œuvre informatique

Recherche dans une séquence en langage naturel

- Pour chaque élément de la séquence à partir du premier :
 - Si cet élément est l'élément cherché, on a terminé
 - Sinon, on continue avec l'élément suivant
- S'il n'y a plus d'éléments et on n'a pas trouvé ce qu'on cherchait, alors il n'est pas là

Recherche dans une séquence en pseudocode

```
fonction chercher(élément, séquence)
  n := longueur(séquence)
  i := 0
  tant que i < n faire
    si séquence[i] = élément alors
      retourner i
    fin si
    i := i + 1
  fin tant que
  retourner -1
fin fonction
```

Recherche dans une séquence en Python 🐍

```
def chercher(element, sequence):  
    n = len(sequence)  
    i = 0  
    while i < n:  
        if sequence[i] == element:  
            return i  
        i = i + 1  
    return -1
```

Indentation du (pseudo)code

```
fonction chercher(élément, séquence)
n := longueur(séquence)
i := 0
tant que i < n faire
  si séquence[i] = élément alors
    retourner i
  fin si
  i := i + 1
fin tant que
retourner -1
fin fonction
```



Indentation du (pseudo)code

```
fonction chercher(élément, séquence)
  n := longueur(séquence)
  i := 0
  tant que i < n faire
    si séquence[i] = élément alors
      retourner i
    fin si
    i := i + 1
  fin tant que
  retourner -1
fin fonction
```



Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```


Exécution d'un algorithme et affectations des variables

fonction somme(n)

s := 0

i := 1

tant que $i \leq n$ **faire**

s := s + i

i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2

Exécution d'un algorithme et affectations des variables

fonction somme(n)

 s := 0

 i := 1



tant que $i \leq n$ **faire**

 s := s + i

 i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2

Exécution d'un algorithme et affectations des variables

fonction somme(n)

s := 0

i := 1



tant que $i \leq n$ **faire**



s := s + i

i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3

Exécution d'un algorithme et affectations des variables

fonction somme(n)

 s := 0

 i := 1

tant que $i \leq n$ **faire**

 s := s + i

 i := i + 1

fin tant que

retourner s

fin fonction



somme(5)

n	s	i
5	0	1
	1	2
	3	3

Exécution d'un algorithme et affectations des variables

fonction somme(n)

 s := 0

 i := 1



tant que $i \leq n$ **faire**

 s := s + i

 i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2
	3	3

Exécution d'un algorithme et affectations des variables

fonction somme(n)

s := 0

i := 1



tant que $i \leq n$ **faire**



s := s + i

i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2
	3	3

Exécution d'un algorithme et affectations des variables

fonction somme(n)

s := 0

i := 1

tant que $i \leq n$ **faire**

s := s + i

i := i + 1

fin tant que

retourner s

fin fonction



somme(5)

n	s	i
5	0	1
	1	2
	3	3

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que i ≤ n faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4

Exécution d'un algorithme et affectations des variables

fonction somme(n)

s := 0

i := 1



tant que $i \leq n$ **faire**

s := s + i

i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4

Exécution d'un algorithme et affectations des variables

fonction somme(n)

s := 0

i := 1



tant que $i \leq n$ **faire**



s := s + i

i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5

Exécution d'un algorithme et affectations des variables

fonction somme(n)

 s := 0

 i := 1



tant que $i \leq n$ **faire**

 s := s + i

 i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5

Exécution d'un algorithme et affectations des variables

fonction somme(n)

s := 0

i := 1



tant que $i \leq n$ **faire**



s := s + i

i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que i ≤ n faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que i ≤ n faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que  $i \leq n$  faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6

Exécution d'un algorithme et affectations des variables

fonction somme(n)

 s := 0

 i := 1



tant que $i \leq n$ **faire**

 s := s + i

 i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6

Exécution d'un algorithme et affectations des variables

fonction somme(n)

s := 0

i := 1



tant que $i \leq n$ **faire**



s := s + i

i := i + 1

fin tant que

retourner s

fin fonction

somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que i ≤ n faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```



somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6

Exécution d'un algorithme et affectations des variables

```
fonction somme(n)  
  s := 0  
  i := 1  
  tant que i ≤ n faire  
    s := s + i  
    i := i + 1  
  fin tant que  
  retourner s  
fin fonction
```

somme(5)

n	s	i
5	0	1
	1	2
	3	3
	6	4
	10	5
	15	6



Structures de contrôle

instruction₁
instruction₂
...
instruction_n

si *condition* **alors**
 instructions
sinon
 d'autres instructions
fin si

tant que *condition* **faire**
 instructions
fin tant que

Séquence d'instructions

fonction discriminant(a, b, c)

(calcul du Δ de l'équation $ax^2 + bx + c = 0$)

d := b²

e := 4ac

Δ := d - e

retourner Δ

fin fonction

Conditions

fonction solutions(a, b, c)

(calcul des solutions de $ax^2 + bx + c = 0$)

$\Delta := b^2 - 4ac$ *(discriminant)*

si $\Delta > 0$ **alors**

$x_1 := (-b + \sqrt{\Delta}) / 2a$

$x_2 := (-b - \sqrt{\Delta}) / 2a$

retourner $\{x_1, x_2\}$ *(deux solutions)*

sinon si $\Delta = 0$ **alors**

retourner $\{-b / 2a\}$ *(une solution)*

sinon

retourner \emptyset *(aucune solution)*

fin si

fin fonction

Itération

fonction puissance(a, n)

(calcul de la puissance a^n avec $n \geq 0$)

si $n < 0$ **alors**

erreur « exposant négatif »

fin si

$p := 1$

$m := n$

tant que $m > 0$ **faire**

$p := p \times a$

$m := m - 1$

fin tant que

retourner p

fin fonction

Composition des structures de contrôle

```
si x < 10 alors
  si y < 30 alors
    x = x + y
  sinon
    y = y - x
  fin si
sinon
  tant que x < 20 faire
    tant que y > 40 faire
      y := y - 1
      x := x + 1
    fin tant que
  fin tant que
fin si
```

Appeler une fonction

fonction solutions(a, b, c)

(calcul des solutions de $ax^2 + bx + c = 0$)

$\Delta := \text{discriminant}(a, b, c)$

si $\Delta > 0$ **alors**

$x_1 := (-b + \sqrt{\Delta}) / 2a$

$x_2 := (-b - \sqrt{\Delta}) / 2a$

retourner $\{x_1, x_2\}$

sinon si $\Delta = 0$ **alors**

retourner $\{-b / 2a\}$

sinon

retourner \emptyset

fin si

fin fonction

Appeler une fonction

fonction solutions(a, b, c)

(calcul des solutions de $ax^2 + bx + c = 0$)

$\Delta := \text{discriminant}(a, b, c)$

si $\Delta > 0$ **alors**

$x_1 := (-b + \sqrt{\Delta}) / 2a$

$x_2 := (-b - \sqrt{\Delta}) / 2a$

retourner $\{x_1, x_2\}$

sinon si $\Delta = 0$ **alors**

retourner $\{-b / 2a\}$

sinon

retourner \emptyset

fin si

fin fonction

fonction discriminant(a, b, c)

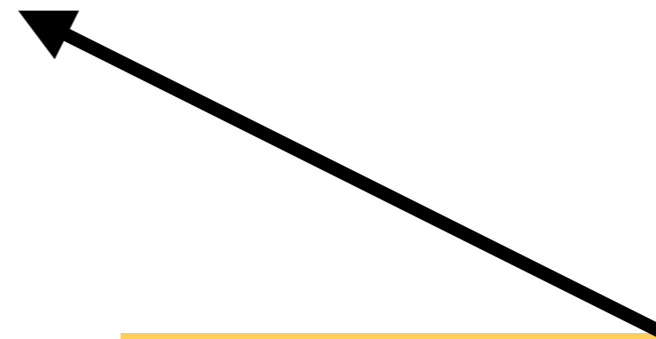
$d := b^2$

$e := 4ac$

$\Delta := d - e$

retourner Δ

fin fonction



Types de données

fonction puissance(a : réel, n : entier) : réel

(calcul de la puissance a^n avec $n \geq 0$)

si $n < 0$ **alors**

erreur « exposant négatif »

fin si

p : réel := 1

m : entier := n

tant que $m > 0$ **faire**

 p := p × a

 m := m - 1

fin tant que

retourner p

fin fonction

Propriétés souhaitées pour un algorithme

- **Terminaison** : l'algorithme termine en un temps fini
- **Correction** : l'algorithme calcule le bon résultat
- **Efficacité** : l'algorithme est aussi rapide que possible (ou au moins aussi rapide que nécessaire)

Somme des n premiers entiers

```
fonction somme(n)
  s := 0
  i := 1
  tant que i ≤ n faire
    s := s + i
    i := i + 1
  fin tant que
  retourner s
fin fonction
```

Terminaison ?

Correction ?

Efficacité ?

Somme des n premiers entiers

```
fonction somme(n)
  s := 0
  i := 1
  tant que i ≤ n faire
    s := s + i
    i := i + 1
  fin tant que
  retourner s
fin fonction
```

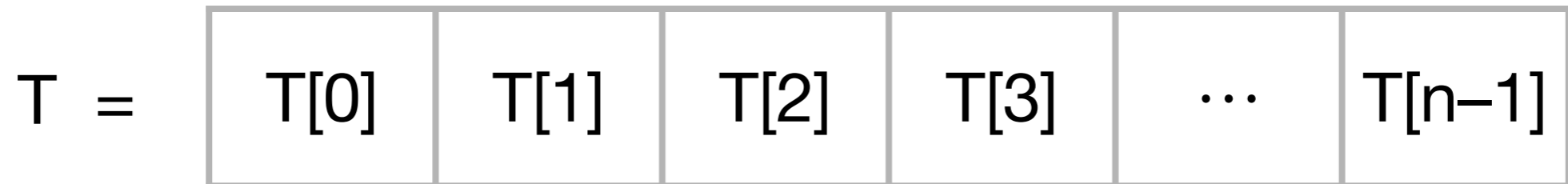
```
fonction somme-à-la-Gauss(n)
  retourner n (n + 1) / 2
fin fonction
```

Terminaison ?

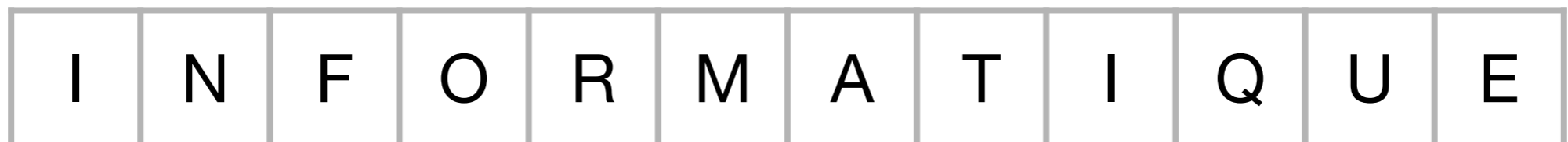
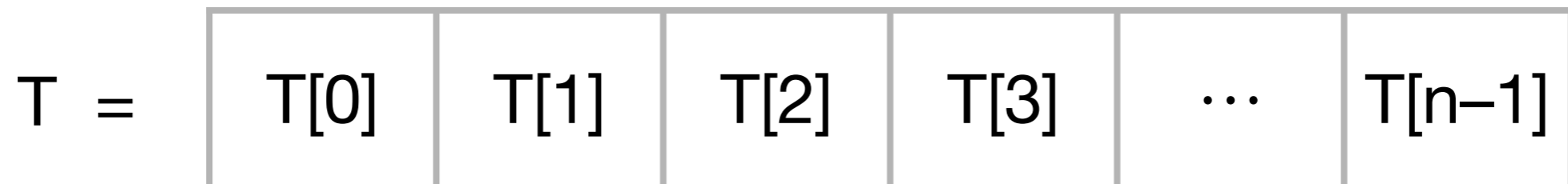
Correction ?

Efficacité ?

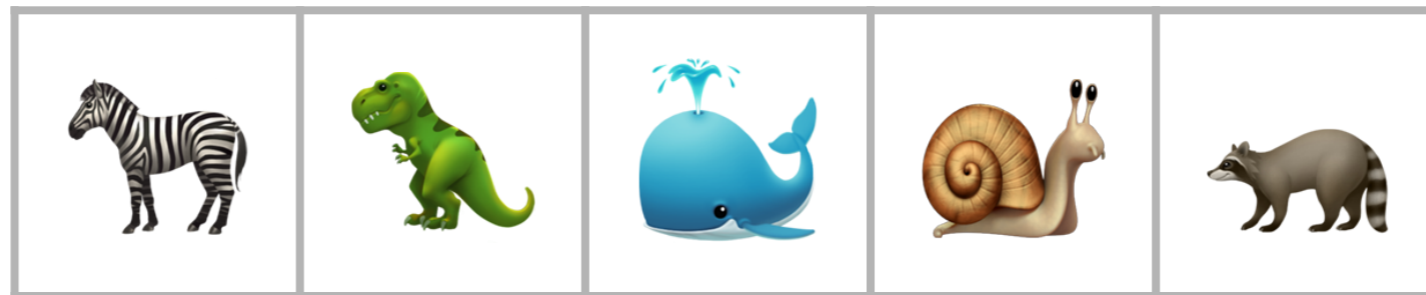
Structures de données : les tableaux



Structures de données : les tableaux



Tableaux



<table border="1"><tbody><tr><td>3</td></tr><tr><td>5</td></tr><tr><td>7</td></tr></tbody></table>	3	5	7	<table border="1"><tbody><tr><td>1</td></tr><tr><td>2</td></tr></tbody></table>	1	2	<table border="1"><tbody><tr><td>42</td></tr><tr><td>5</td></tr></tbody></table>	42	5	<table border="1"><tbody><tr><td>13</td></tr></tbody></table>	13	<table border="1"><tbody><tr><td>1</td></tr><tr><td>3</td></tr><tr><td>2</td></tr></tbody></table>	1	3	2
3															
5															
7															
1															
2															
42															
5															
13															
1															
3															
2															

Parcourir un tableau

```
procedure parcours(T)
  n := longueur(T)
  i := 0
  tant que i < n faire
    écrire T[i]
    i := i + 1
  fin tant que
fin procedure
```