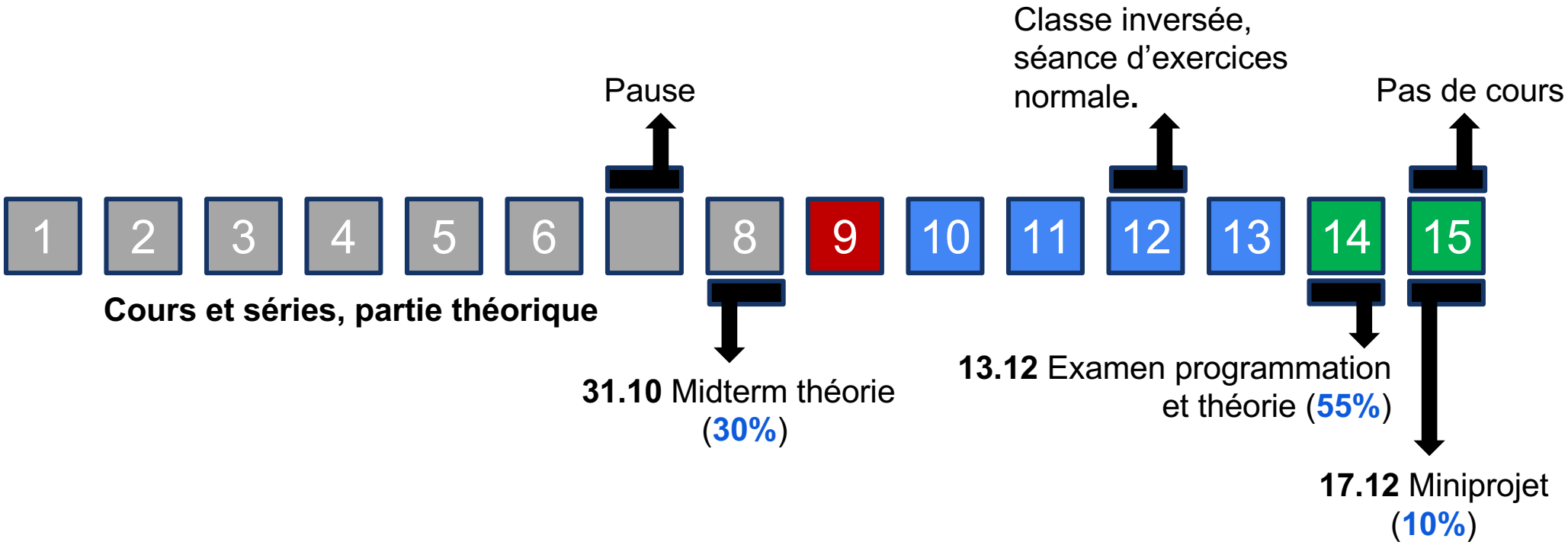
Information, Calcul et Communication

CS-119(g) ICC – Théorie
Semaine 9

Rafael Pires
rafael.pires@epfl.ch

Lausanne, 07.11.2025

Programme du cours



Programme du cours

Calcul

- Algorithmes
- Complexité
- Conception d'algorithmes
- Calculabilité
- **Circuits, architecture**

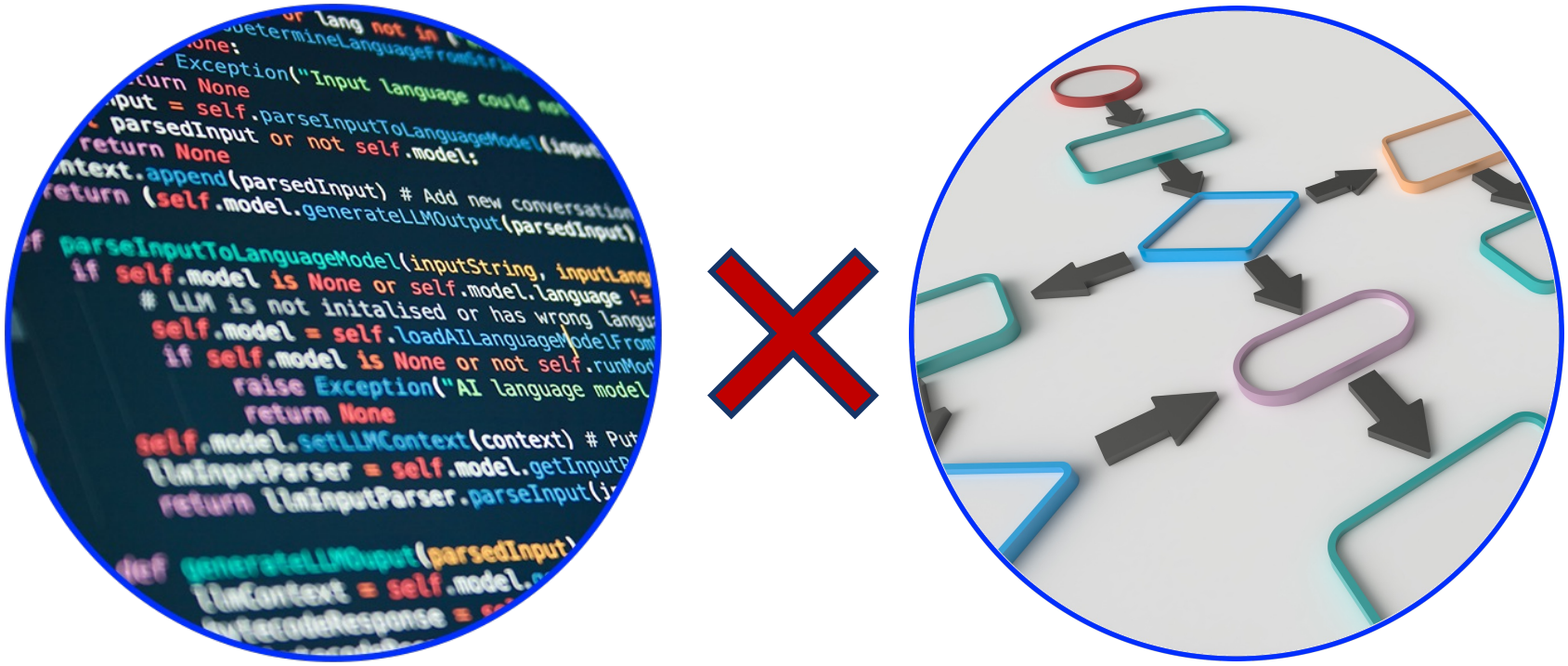
Information

- **Représentation de nombres**
- **Echantillonnage et reconstruction de signaux**
- **Entropie**
- **Compression**

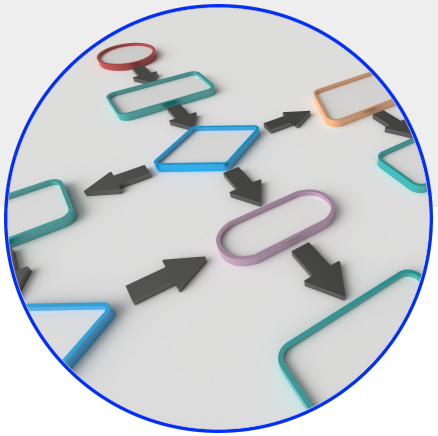
Communication

- **Réseau**
- **Cryptographie**

Programme vs. algorithme



Qu’est-ce qu’un algorithme ?



- Un algorithme n’est **pas** un programme.
- Un **algorithme** est la description des étapes élémentaires menant à la résolution d’un problème; c’est donc la **description conceptuelle** d’un programme.
- Un **programme** est **l’implémentation** d’un algorithme dans un langage donné et dans un système particulier.

Exemple 1 : calcul du modulo 3 d'un grand nombre

76321 | 3

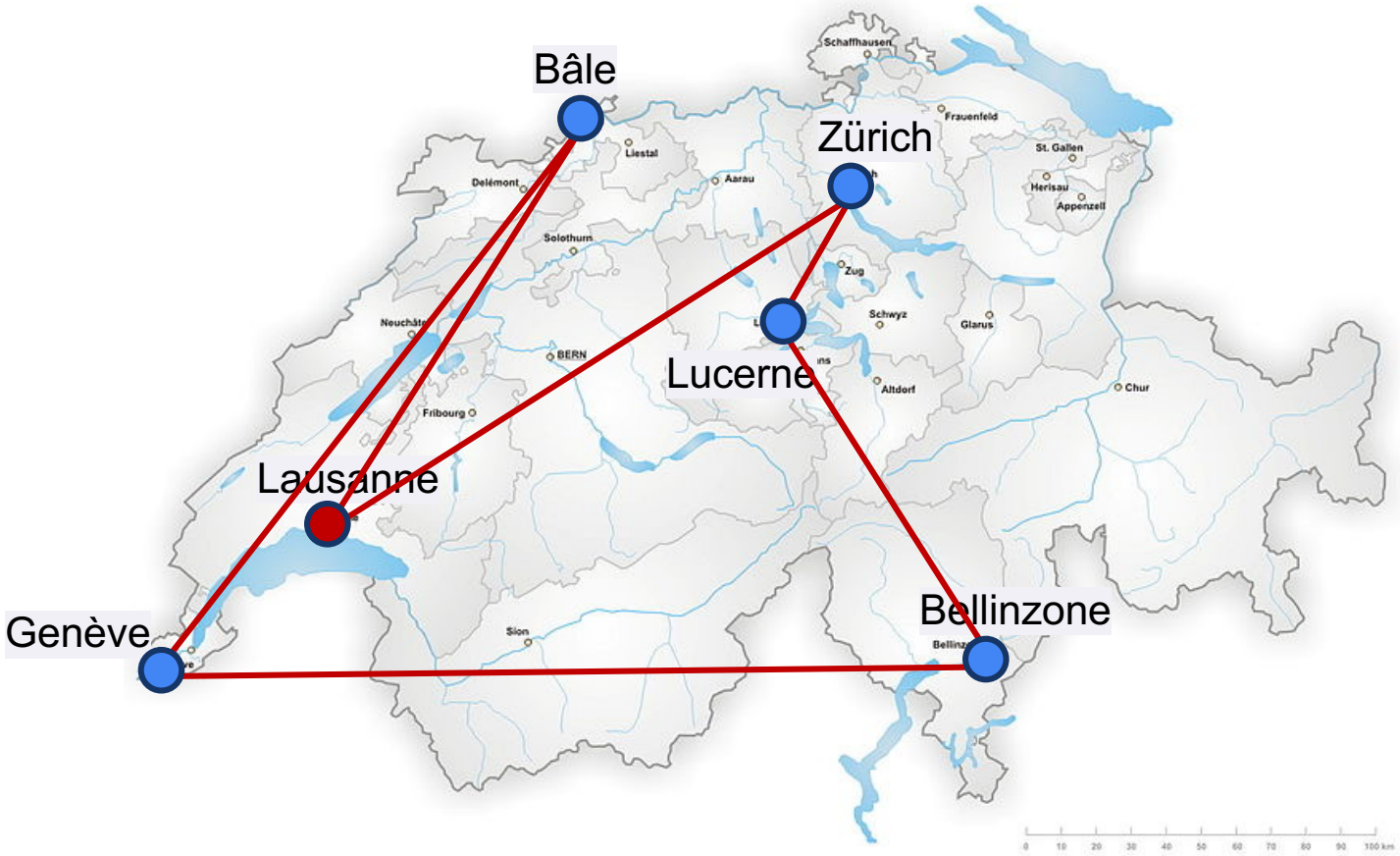
Rappel :

Modulo : le reste r de la division d'un entier a par un entier b non nul.
$a \bmod b = r$, si $a = q.b + r$ et $0 \leq r < |b|$

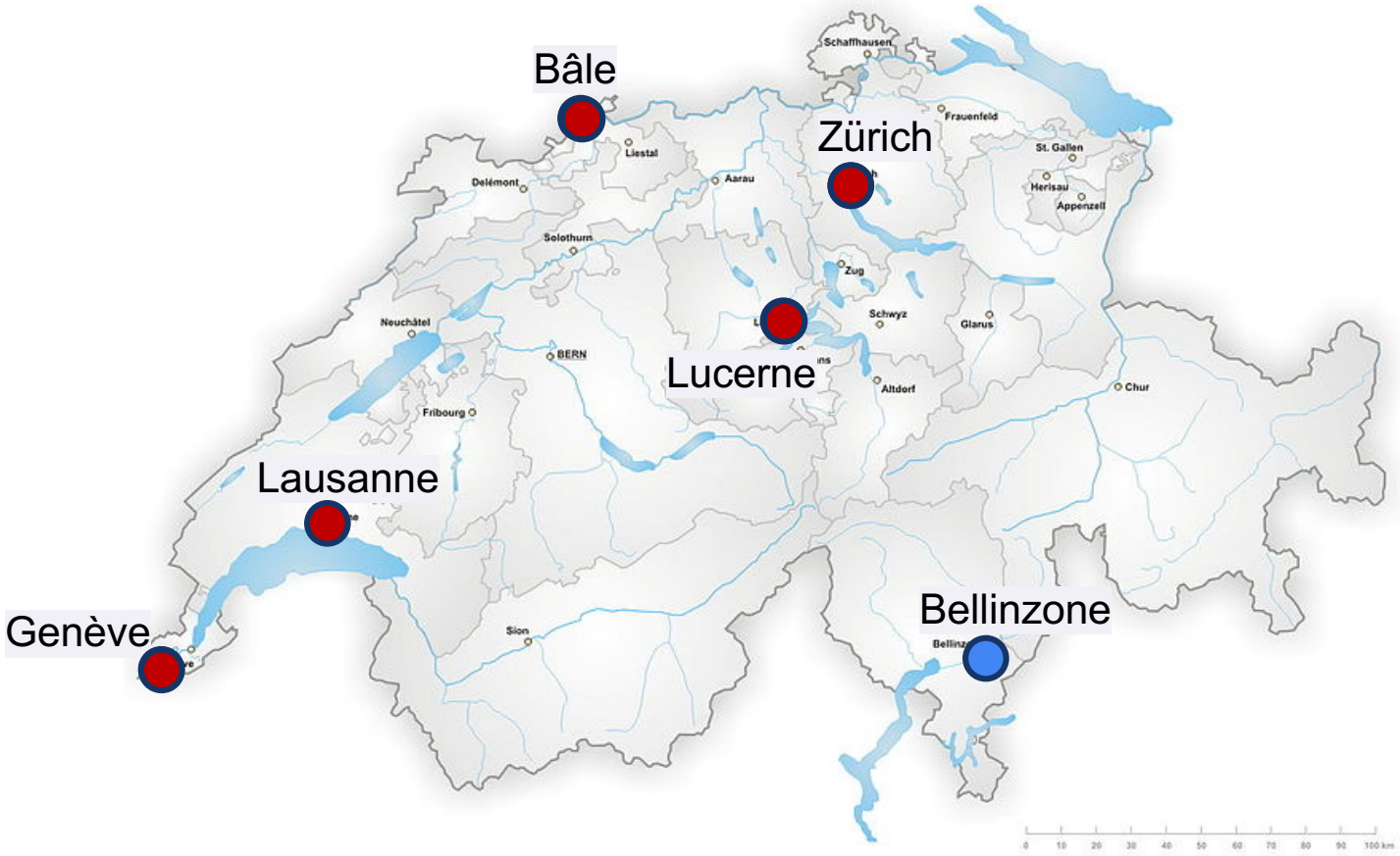
Exemple 2 : recherche du minimum dans une liste

$$\begin{cases} L = (13, 47, 18, 15, 11, 19, 46, 18, 15) \\ n = 9 \end{cases}$$

Exemple 3 : problème du voyageur de commerce



Exemple 3 : problème du voyageur de commerce



Algorithmes : ingrédients de base

Données

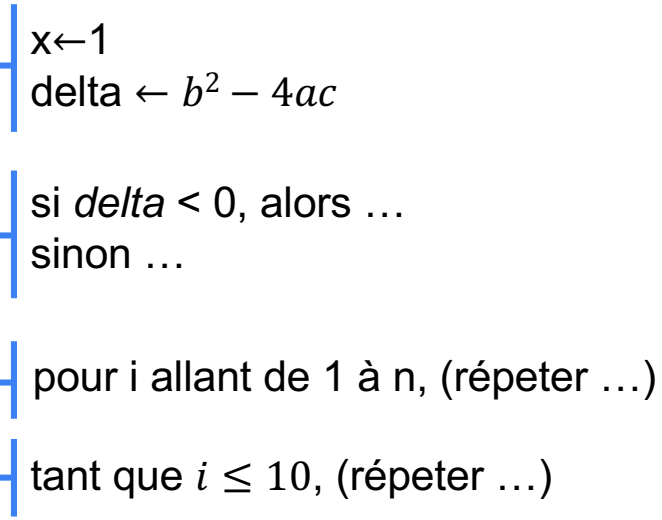
- Entrées
- Sorties
- Variables internes

Traitement

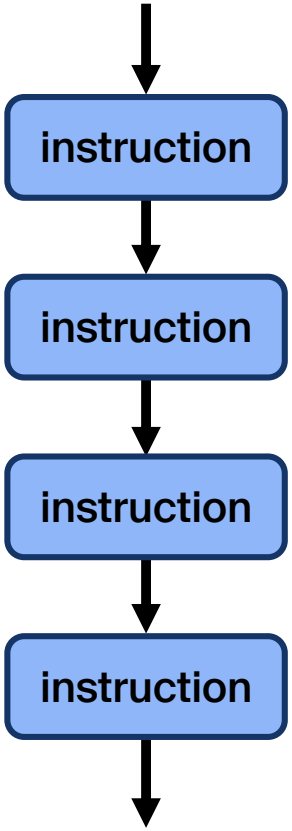
- Affectations
- Structures de contrôle
  - Branchements conditionnels (tests)
  - Itérations (boucles)
  - Boucles conditionnelles

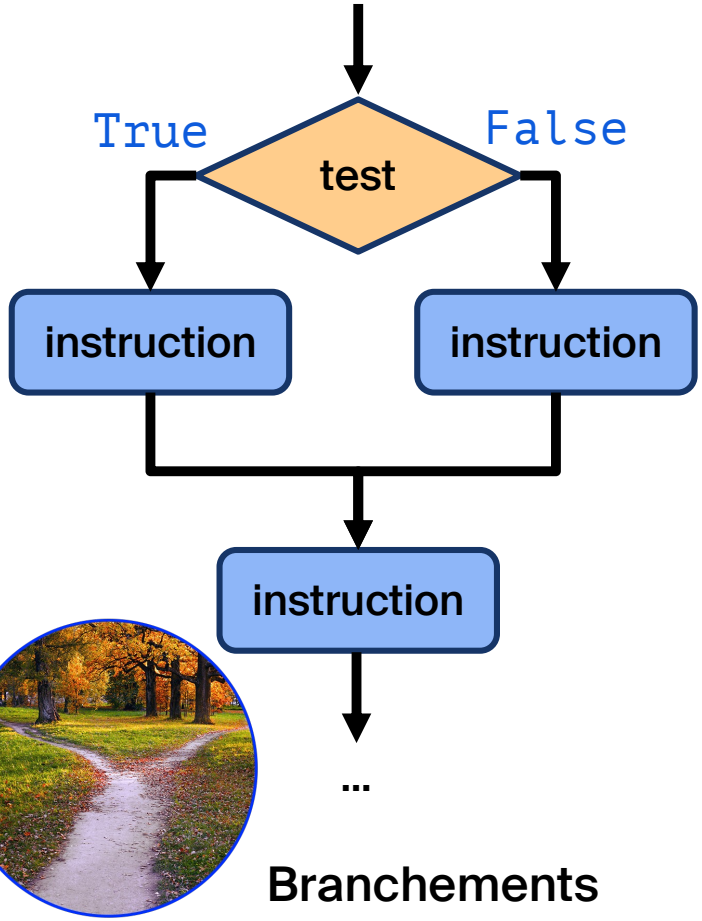
Algorithmes : instructions

- Affectations
- Structures de contrôle
  - Branchements conditionnels (tests)
  - Itérations (boucles)
  - Boucles conditionnelles

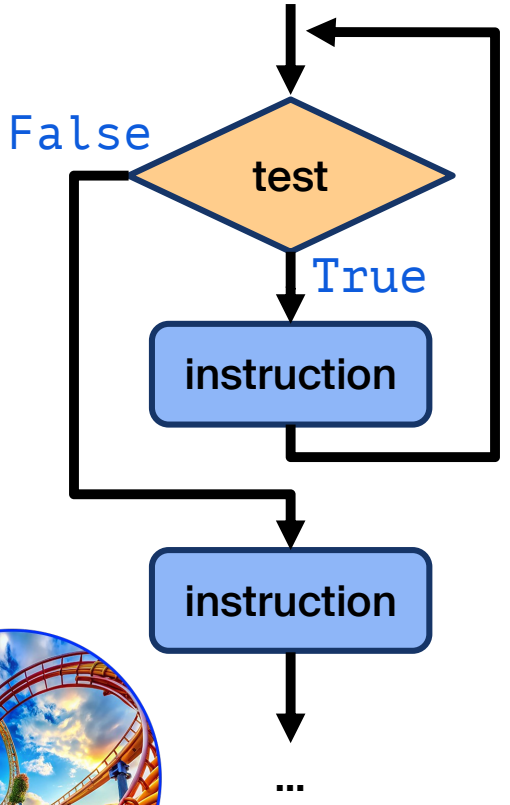


Structures de contrôle





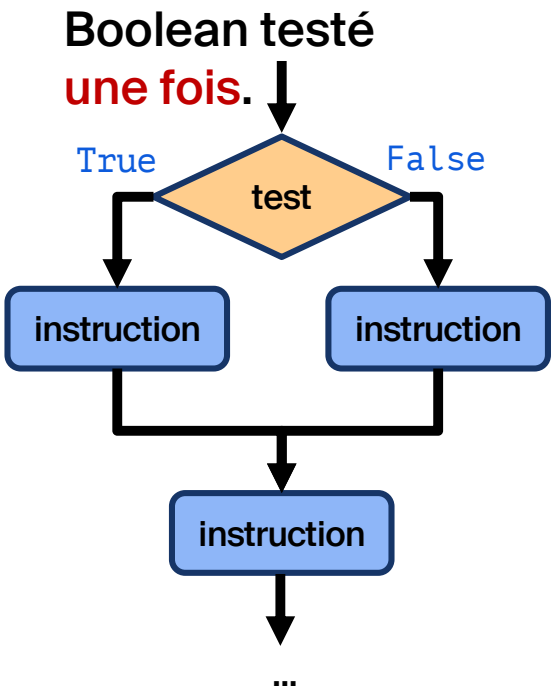
Branchements

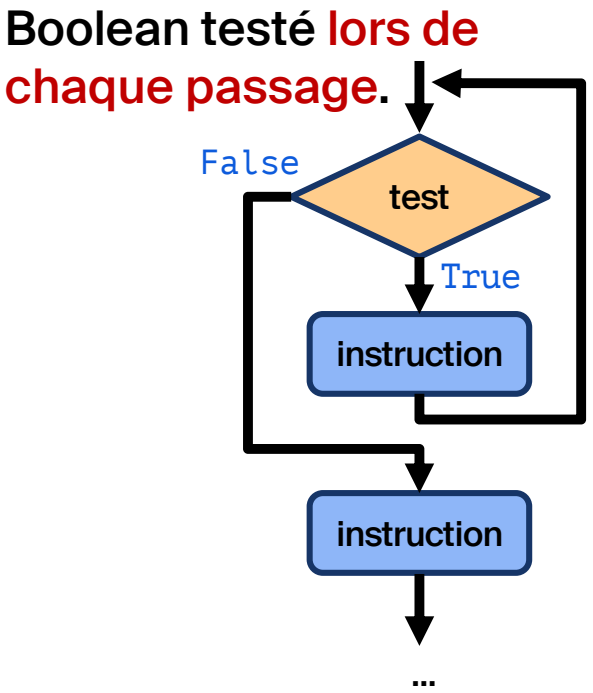


Boucles

Le test

- **Chaque condition nécessite une valeur booléenne pour orienter la suite du traitement**
- Soit vrai (`True`), soit faux (`False`)





Ingrédients de base

Données

- Entrées
- Sorties
- Variables internes

Instructions

- Affectations
- Structures de contrôle
  - Branchements conditionnels (tests)
  - Itérations (boucles)
  - Boucles conditionnelles

L = (13, 47, 18, 15, 11, 19, 46, 18, 15)
n = 9

1. On considère le premier nombre de la liste le '`minimum`'
2. On le compare au 2ème nombre, le '`suivant`'
3. Si `suivant` < `minimum`, alors `minimum` ← `suivant`
4. Sinon, on continue, c'est à dire, `suivant` ← celui d'après
5. On revient à l'étape 3 jusqu'à la fin de la liste
6. Le résultat est la valeur de '`minimum`'

Pseudo-code

L = (13, 47, 18, 15, 11, 19, 46, 18, 15)
n = 9

1. On considère le premier nombre de la liste le '**`minimum`**'
2. On le compare au 2ème nombre, le '**`suivant`**'
3. Si **`suivant`** < **`minimum`**, alors **`minimum`** ← **`suivant`**
4. Sinon, on continue, c'est à dire, **`suivant`** ← celui d'après
5. On revient à l'étape 3 jusqu'à la fin de la liste
6. Le résultat est la valeur de '**`minimum`**'

Valeur minimale

entrée : liste ***L*** de nombres entiers, de taile ***n***
sortie : la (ou une des) valeur(s) minimale(s) de la liste

```
min ← L(1)
Pour i allant de 2 à n
        Si L(i) < min
                min ← L(i)
Sortir : min
```

Attention !

Boucles

Indices et comparateur d'inégalité

ICC-T : vendredi

≠

ICC-P : lundi

Itération

pour i allant de 1 à 10,
(répeter …)

≡

Boucle conditionnelle

$i \leftarrow 1$
tant que $i \leq 10$
(répeter …)
$i \leftarrow i + 1$

Itération

```
for (i = 0; i < 10; ++i) {
    // (répeter …)
}
```

≡

Boucle conditionnelle

```
i = 0;
while (i < 10) {
    // (répeter …)
    i = i + 1;
}
```

pour i allant de 1 à ***n***,
(répeter …)

n – 1 + 1 = n

Nombre de itérations

=

```
for (i = 0; i < n; ++i) {
    // (répeter …)
}
```

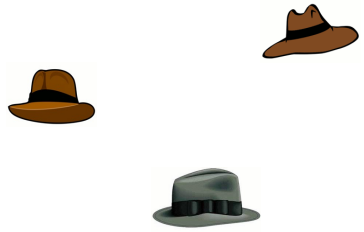
n – 0 = n

Comparaison tous contre tous

- **Question**
  - Est-ce que tous les objets visibles sur cette photo sont différents les uns des autres ?

Tous les 3 différents ?

- **Problème à résoudre :**
  - Parmi une liste de 3 objets, identifier si ceux-ci sont tous différents les uns des autres.



Tous les 3 différents

entrée : liste ***L*** de 3 objets
sortie : valeur binaire oui/non

s ← ***oui***
Pour ***i*** *allant de 1 à 3 :*
    Pour ***k*** *allant de 1 à 3* **:**
        Si ***L***(***i***) = ***L***(***k***) et ***i*** ≠ ***k***, alors ***s*** ← ***non***
Sortir : ***s***

Tous les 3 différents

entrée : liste ***L*** de 3 objets
sortie : valeur binaire oui/non

s ← ***oui***
Si ***L***(1) = ***L***(2), alors ***s*** ← ***non***
Si ***L***(1) = ***L***(3), alors ***s*** ← ***non***
Si ***L***(2) = ***L***(3), alors ***s*** ← ***non***
Sortir : ***s***

Comparaisons dans une boucle imbriquée

i \ k	1	2	3
1	1,1	1,2	1,3
2	2,1	2,2	2,3
3	3,1	3,2	3,3

Tous différents ?

- **Problème à résoudre :**
  - Parmi une liste de n objets, identifier si ceux-ci sont tous différents les uns des autres.

Tous différents

entrée : liste ***L*** de ***n*** objets
sortie : valeur binaire oui/non

s ← ***oui***
Pour ***i*** *allant de 1 à* ***n****-1 :*
    Pour ***k*** *allant de* ***i****+1 à* ***n*** **:**
        Si ***L***(***i***) = ***L***(***k***), alors ***s*** ← ***non***
Sortir : ***s***

Tous différents ?

- **Problème à résoudre :**
  - Parmi une liste de n objets, identifier si ceux-ci sont tous différents les uns des autres.

Tous différents

entrée : liste ***L*** de ***n*** objets
sortie : valeur binaire oui/non

s ← ***oui***
Pour ***i*** *allant de 1 à* ***n****-1 :*
    Pour ***k*** *allant de* ***i****+1 à* ***n*** **:**
        Si ***L***(***i***) = ***L***(***k***), alors ***s*** ← ***non***
Sortir : ***s***

Algorithme d'Euclide

- L'algorithme d'Euclide utilise une boucle conditionnelle pour trouver le plus grand diviseur commun (pgdc) de deux nombres entiers.

pgdc

entrée : ***a***, ***b***, deux nombres entiers positifs
sortie : pgdc(a, b)

tant que ***b*** ≠ 0
    temp ← ***b***
    b ← ***a*** mod ***b***
    a ← ***temp***
Sortir : ***a***

pgdc(a, b) = pgdc(a-b, b) = pgdc(a-k.b, b) = pgdc(a mod b, b)

Résumé Cours semaine 9 – ICC-T

- Programme vs. Algorithme
- Exemples d’algorithmes
- Ingrédients de base : donnés et instructions
- Boucle imbriquée / conditionnelle

rafael.pires@epfl.ch

Merci