



# **Initiation à Algorithmique**

## **Langage d'application ADA**

MCours.com



# Infos pratiques

Organisation du semestre :

Cours/TDs

Certains exercices de Tds seront programmés sont réalisés en binôme et notés.

Un partiel en janvier.

Bibliographie :

*Algorithmique pour les BTS et IUT « 1. Les bases de la programmation »*, A.Maunoury, K.Ben Sassi

*Le langage ADA « Théorie et Pratique »*, M.Alabau

*The Art of Computer Programming*, Donald E. Knuth, Volumes 1-3.



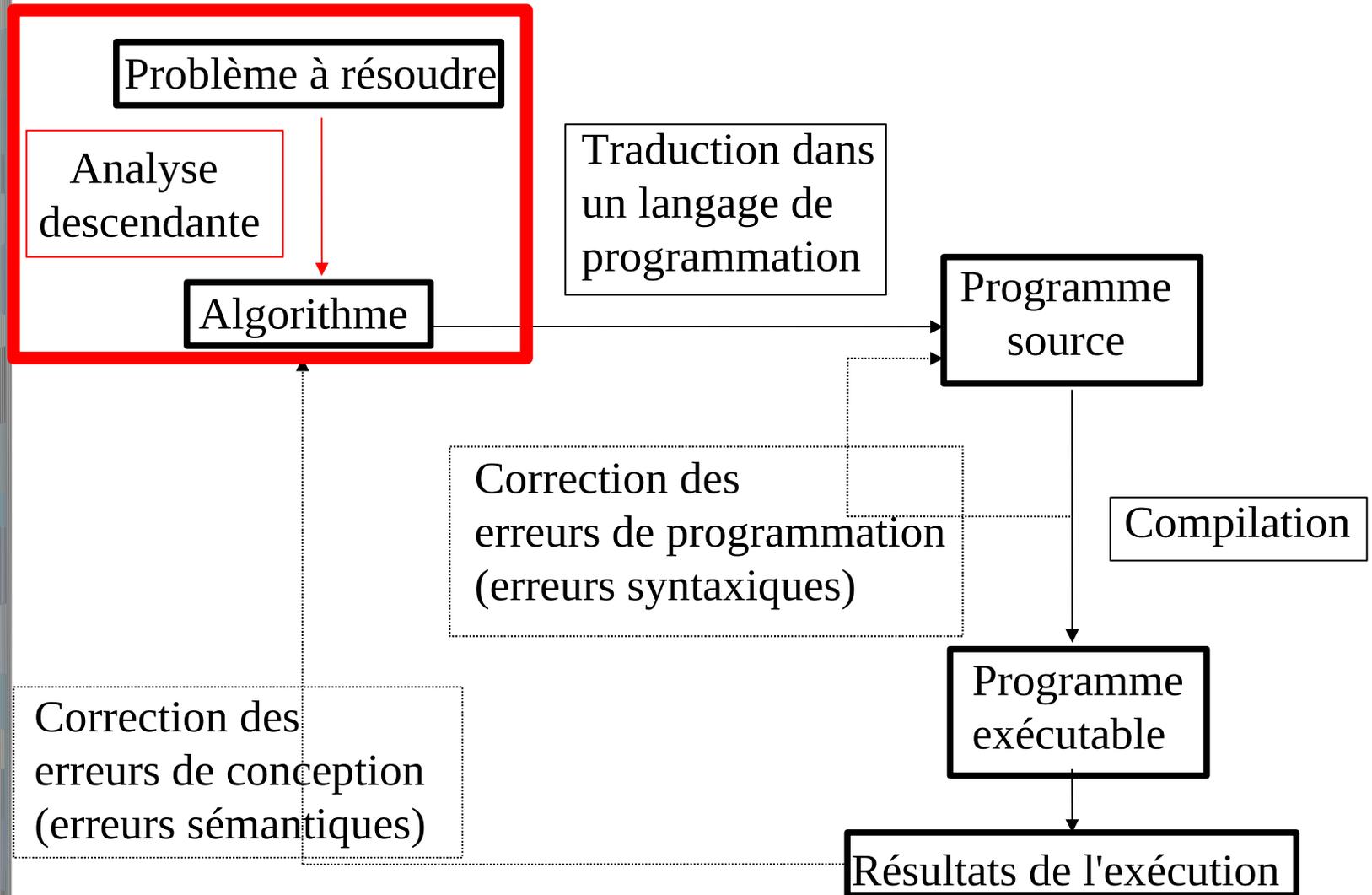
# L'algorithmique

Un **algorithme** est une suite d'instructions qui une fois exécutée, conduit à un résultat donné.

Un **algorithme** est une suite d'actions que devra effectuer un automate pour arriver en un temps fini, à un résultat déterminé à partir d'une situation donnée. La suite d'opérations sera composée d'actions élémentaires appelées **instructions**.

**L'algorithmique** est la logique d'écrire des algorithmes. Pour pouvoir écrire des algorithmes, il faut connaître la *résolution manuelle du problème*, connaître *les capacités de l'ordinateur en terme d'actions élémentaires* qu'il peut assurer, et la *logique d'exécution des instructions*.

# Résolution d'un problème



# Exemple culinaire

1. Comprendre l'énoncé du problème

Fabriquer un brownie

2. Décomposer le problème en sous-problèmes : on parle **d'analyse descendante**

- 1 Préparer la pâte
- 2 Faire fondre le chocolat
- 3 Mélanger le chocolat et la pâte
- 4 Faire cuire 35 minutes

3. Associer à chaque sous-problème une spécification :

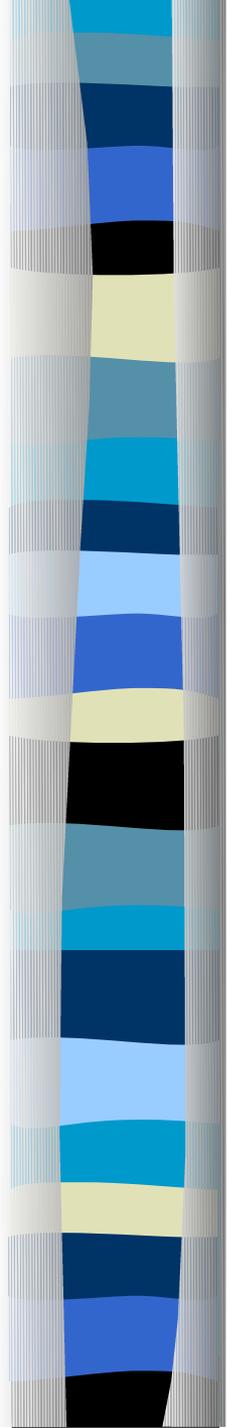
- . les données nécessaires
- . les données résultantes
- . la démarche à suivre pour arriver au résultat en partant d'un ensemble de données

Bol ; tablette de 200g de chocolat ; spatule  
Pâte de chocolat

1. Casser le chocolat en morceaux dans le bol
2. rajouter un peu d'eau
3. mettre le bol 2 mn au micro-onde
4. mélanger le tout avec une spatule afin d'obtenir une pâte lisse

4. **Elaboration de l'algorithme**

**Rassembler toutes les étapes ou instructions et les ordonner (séquentiel)**



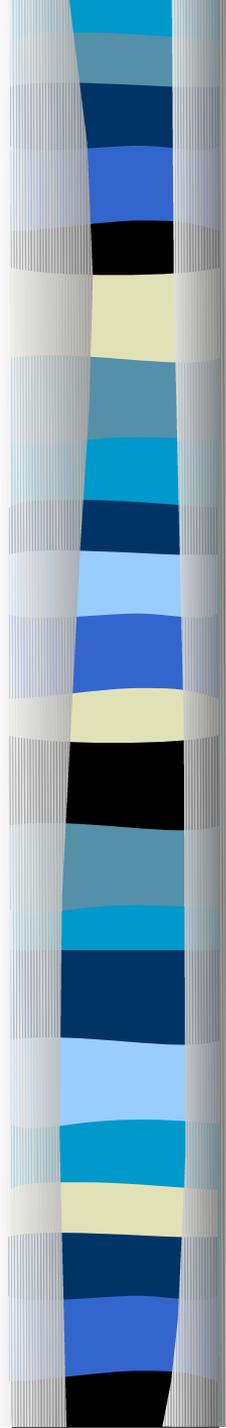
# Analyse descendante : les arbres programmatiques

Les arbres programmatiques, ou le pseudo-code, permettent d'exprimer le résultat de l'analyse sans référence à un langage de programmation en particulier.

Pour celà nous allons utiliser un formalisme pour représenter :

les différents types de données (nécessaires, résultantes)

les actions élémentaires, qui s'enrichira au fur et à mesure du cours.



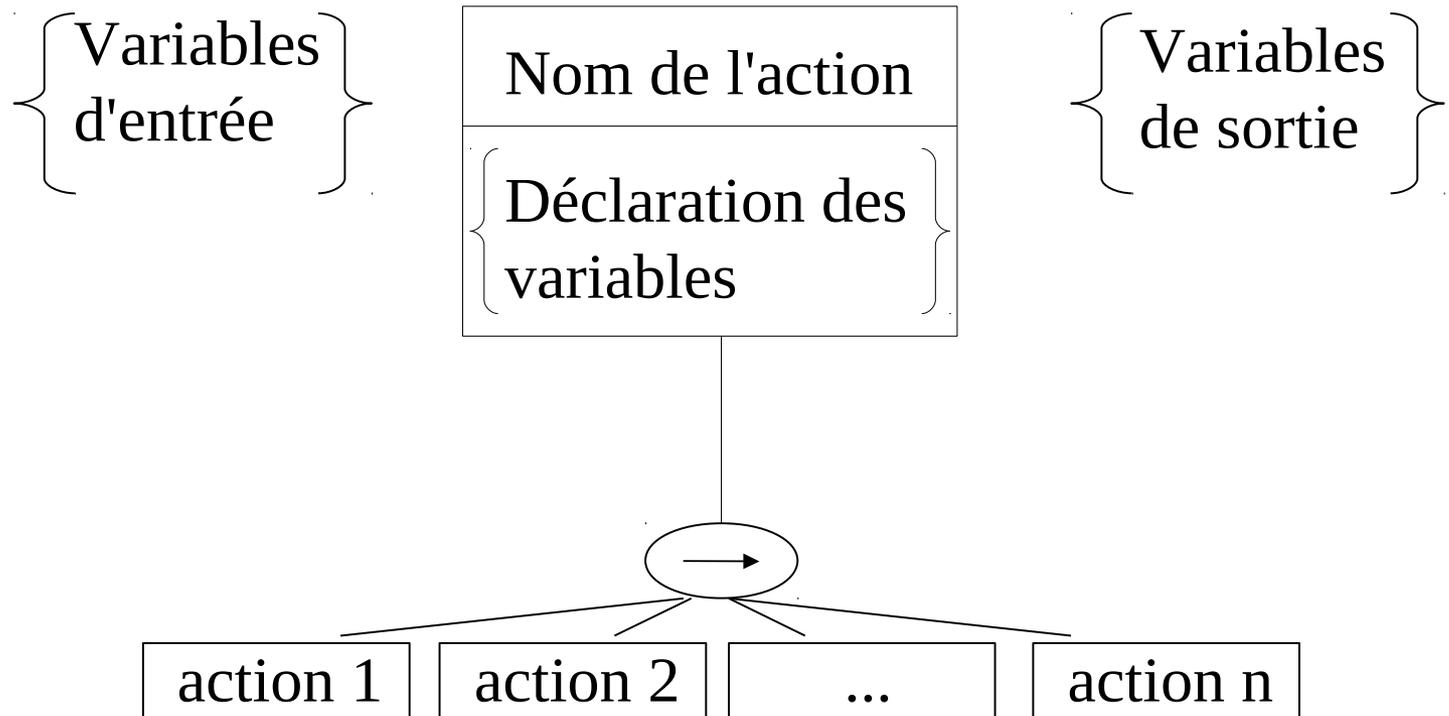
# Construction de l'arbre

## Les données

- Les données nécessaires d'entrée provenant de résultats précédents (pâte de chocolat) seront mises dans les **variables d'entrée**
- Les données résultantes seront appelées les **variables de sortie**
- Les données nécessaires à la réalisation du problème, ou sous-problème, (bol) seront énumérées dans la partie **déclaration des variables**
- Ces 2 types de données seront introduits entre *accolades*
- Le problème, voire les sous-problèmes, sera défini par un nom, le **nom de l'action** qui sera mis dans un *cadre*
- La séquence, la suite séquentielle des actions, sera représentée par une *flèche dans une ellipse*

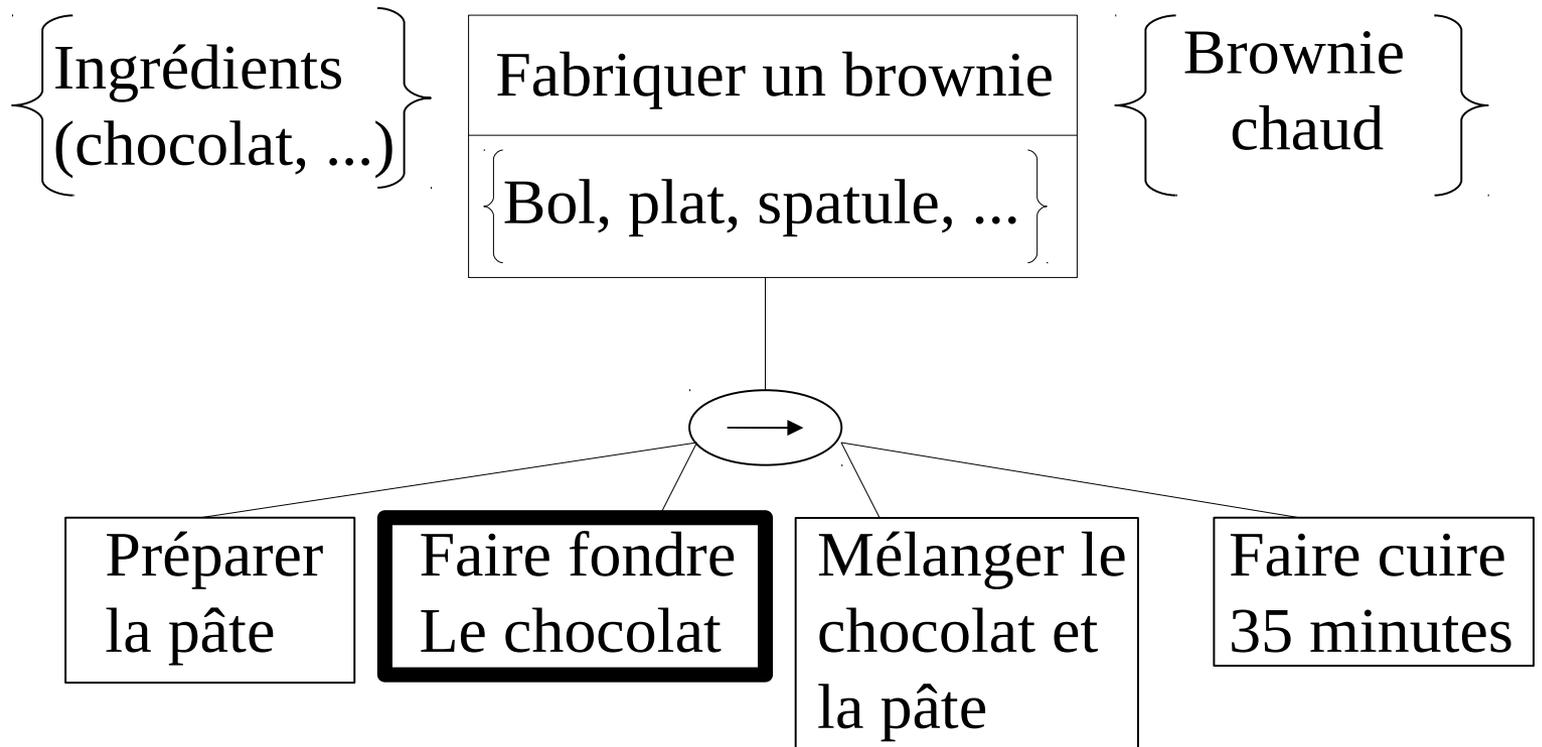
# Construction de l'arbre

## Les données



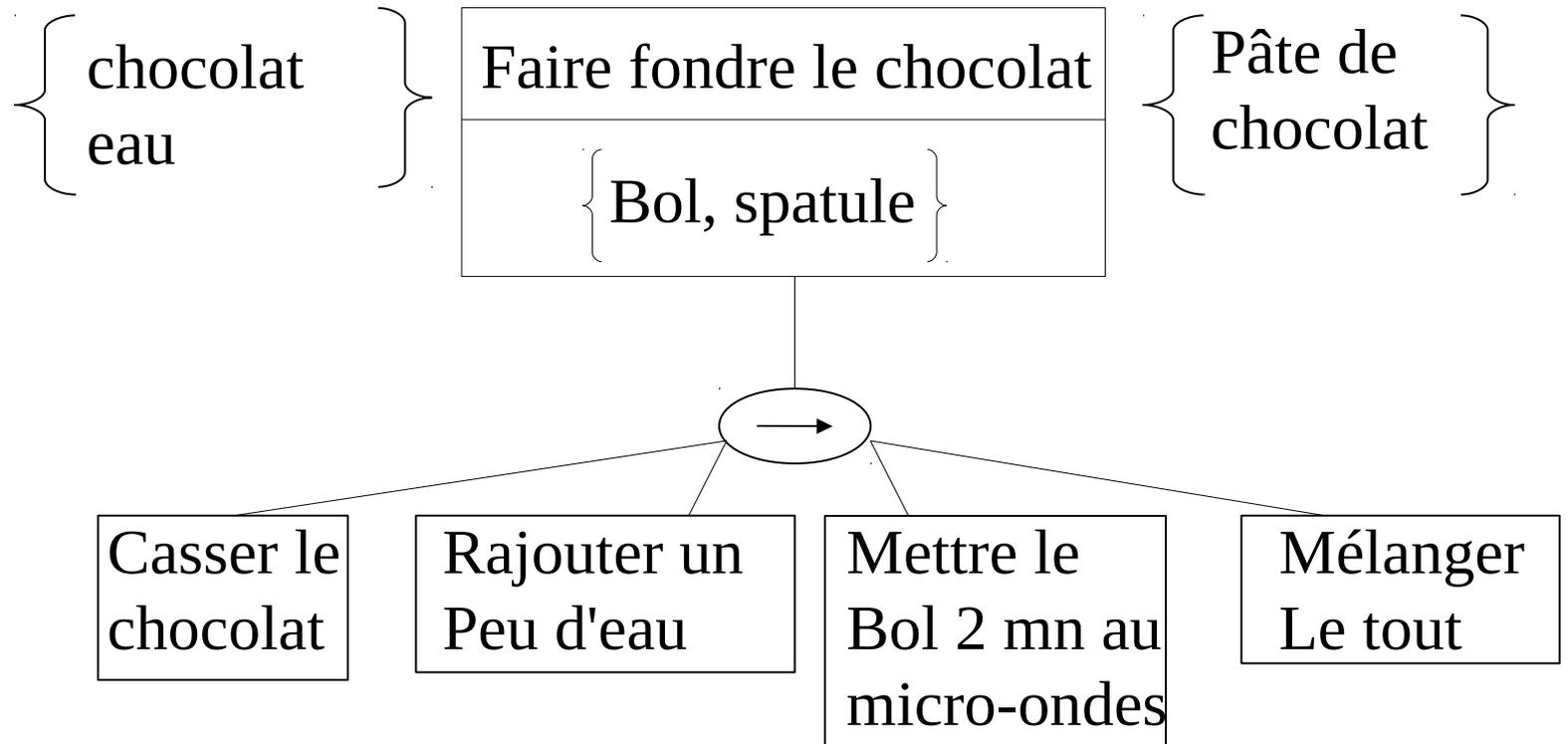
# Les données

## Exemple culinaire



# Les données

## Exemple culinaire



Remarque : ici le micro-ondes est un outil permettant l'opération, comme le '+' permettra d'ajouter 2 nombres



# Algorithme => Ordinateur

L'algorithmique permet de résoudre des problèmes simples comme compliqués, sur un ordinateur.

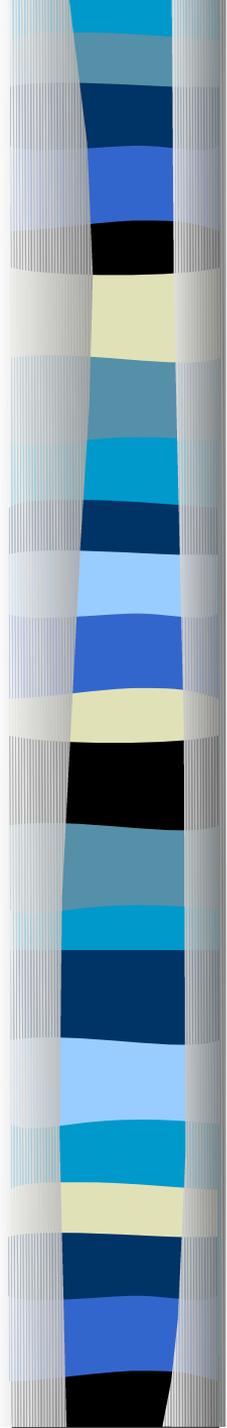
La spécification associée à chaque problème (ou sous-problème) doit se faire avec des outils compréhensibles par l'ordinateur.

Comment fonctionne un ordinateur ?

Il possède un ensemble de cases = **variables**

Il va réaliser des **opérations (opérations mathématiques, affectation, ...)** sur le contenu des cases = **données**

Il possède un ensemble de **mécanismes** qui permettent l'élaboration de l'algorithme (structures de contrôle, répétition, ...)



# Algorithme => Ordinateur

## Exemple culinaire

Ordinateur = maison

Variables = bol, plat

Données = farine, chocolat, eau

Opérations = mélanger, faire fondre, faire cuire

Mécanismes = temps

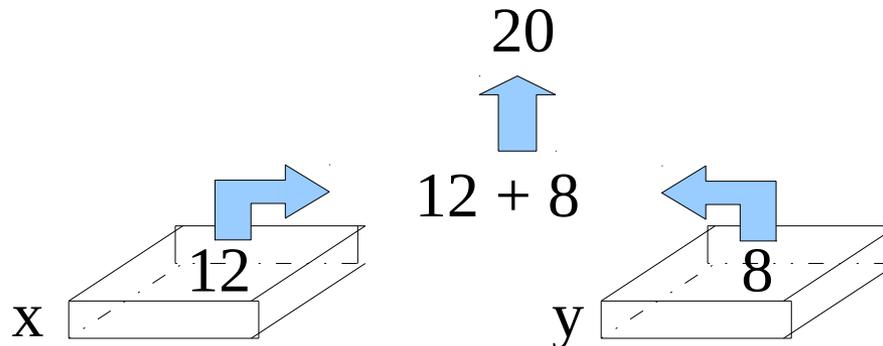
# Algorithme d'un problème simple

$$x+y$$

Problème : donner le résultat de la somme de 2 nombres

Quelle est l'action à faire ?

Ajouter le contenu de  $x$  à celui de  $y$  : *en informatique l'opération  $x+y$  va réaliser cela.*



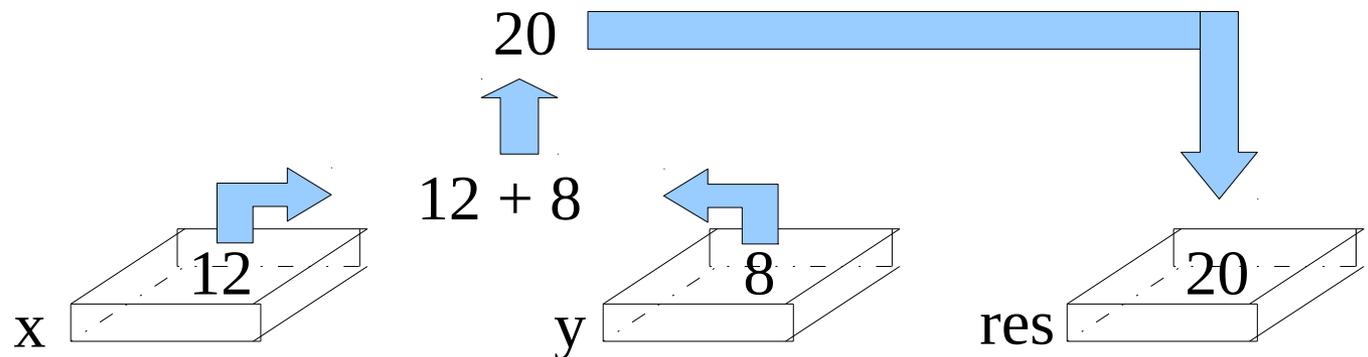
# Algorithme d'un problème simple

## $x+y$

Problème : donner le résultat de la somme de 2 nombres

Quelle est l'action à faire ?

Ajouter le contenu de  $x$  à celui de  $y$  : *en informatique l'opération  $x+y$  va réaliser cela.*



Comment récupérer le résultat de l'opération ?

*En le mettant dans une case*

Comment retrouver la case ? *En lui donnant un nom*



# Algorithme d'un problème simple

$$x+y$$

Comment mettre une donnée (valeur) dans une variable (case) ?

On connaît la valeur de la variable : on fait une ***affectation***

On ne connaît pas la valeur a priori de la variable, on veut que ce soit l'utilisateur qui la rentre au clavier : on va ***lire la valeur de la variable***

# Actions élémentaires

Dans **les variables d'entrée** on nommera toutes les variables que l'on utilisera dans l'algorithme

**Les variables de sortie** contiendront les résultats du problème

Pour réaliser un problème simple, nous avons souvent besoin d'opérations mathématiques de base (+, -, ...), et d'actions élémentaires nous permettant de mettre des valeurs dans nos variables

L'affectation sera symbolisée par ←

La **lecture** permettant de lire la valeur que l'on veut affecter à la variable x sera explicitée par ***lire(x)***

Une dernière action élémentaire est celle qui permet de donner à l'utilisateur la valeur d'une variable x, on dit que l'algorithme **affiche** la valeur contenue dans x et l'action est explicitée par ***afficher(x)***



# Construction de l'arbre problème $x+y$

Pour notre problème, on peut mettre 12 dans la variable nommée  $x$  alors on le formalisera dans notre arbre par :

$x \leftarrow 12$

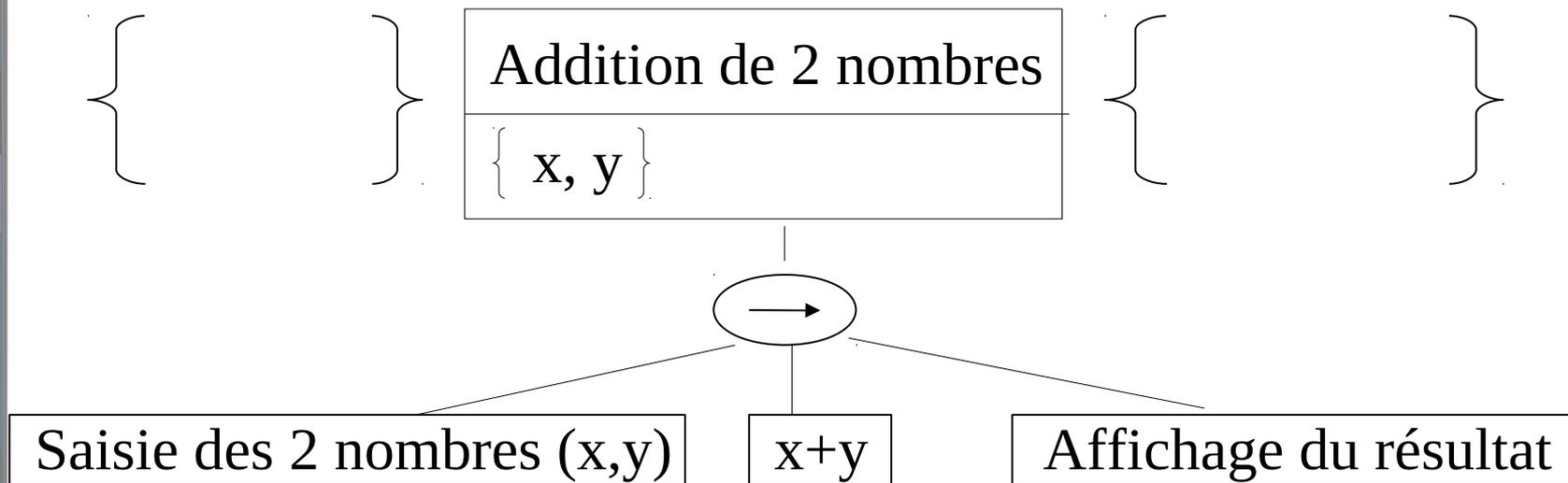
Si on veut que l'algorithme lise la valeur entrée au clavier par l'utilisateur alors on écrira : `lire(x)`

# Construction de l'arbre problème $x+y$

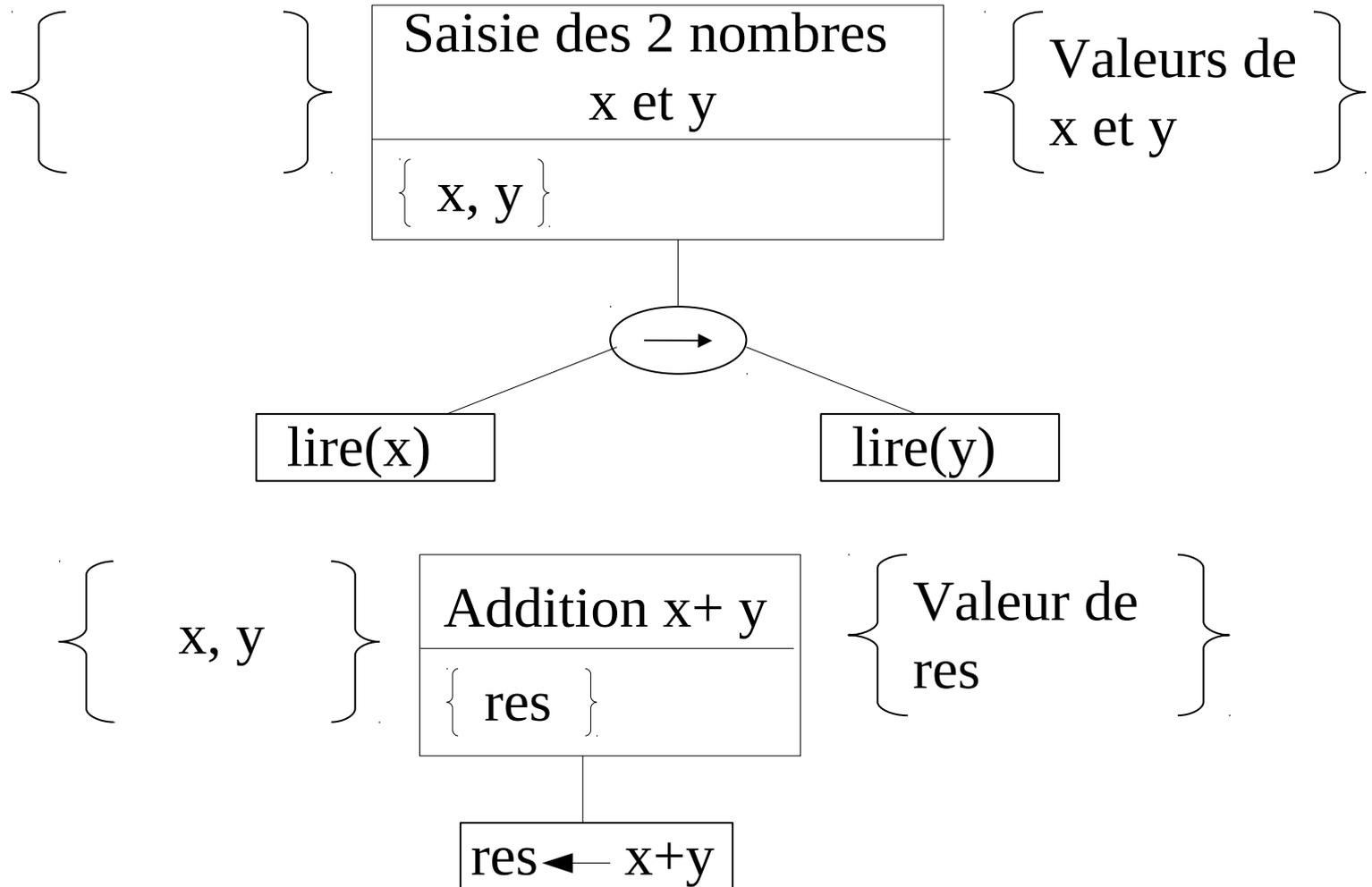
Pour notre problème, on peut mettre 12 dans la variable nommée  $x$  alors on le formalisera dans notre arbre par :

$x \leftarrow 12$

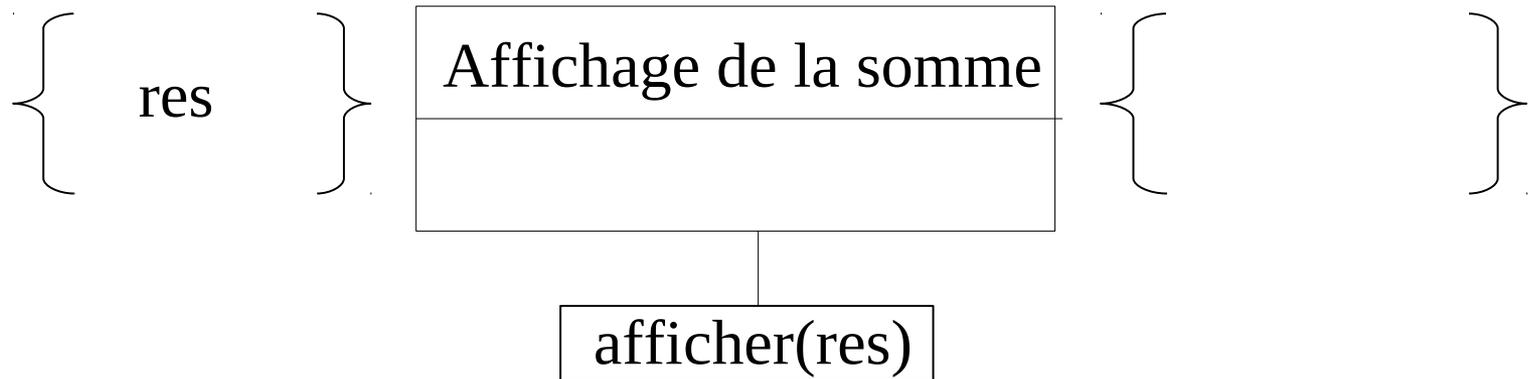
Si on veut que l'algorithme lise la valeur entrée au clavier par l'utilisateur alors on écrira : `lire(x)`



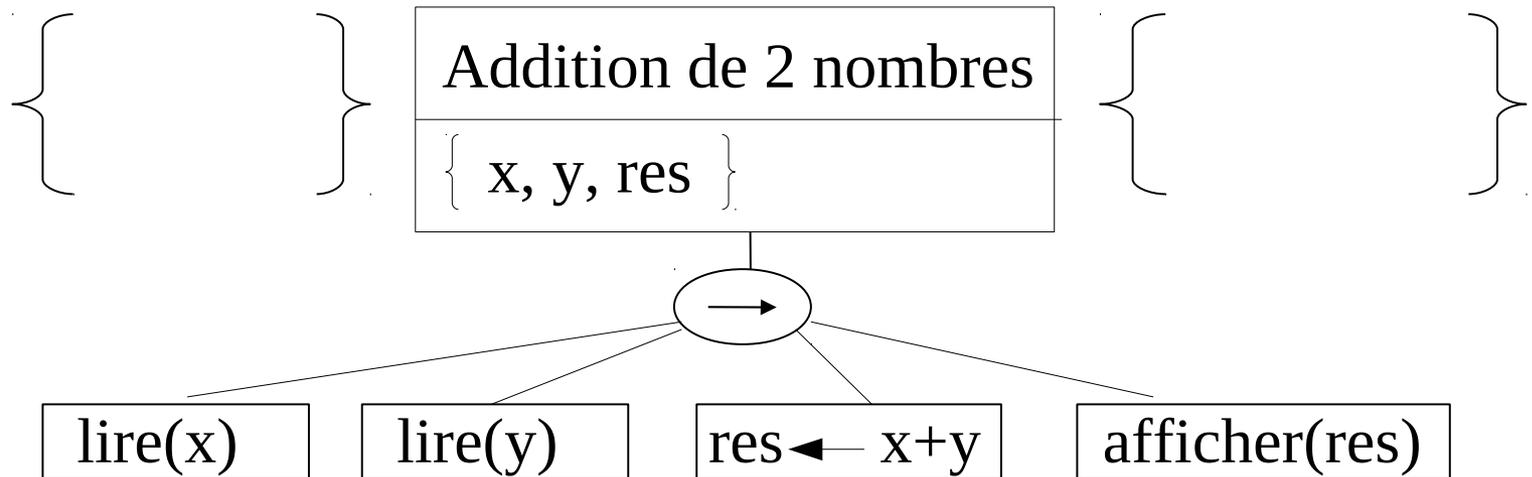
# Construction de l'arbre problème $x+y$

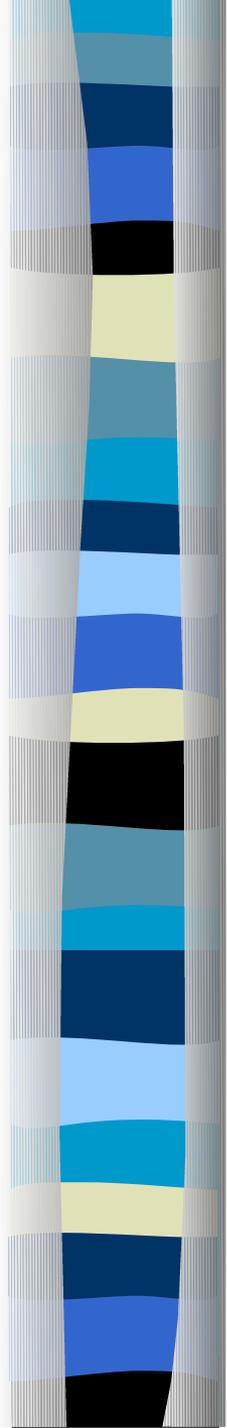


# Construction de l'arbre problème $x+y$



Mais aussi, sans module intermédiaire :





# Exemple

## moyenne de 3 nombres

Comment calcule-t-on la moyenne de 3 nombres ?

On fait la somme des 3 nombres

On divise le résultat de la somme par 3

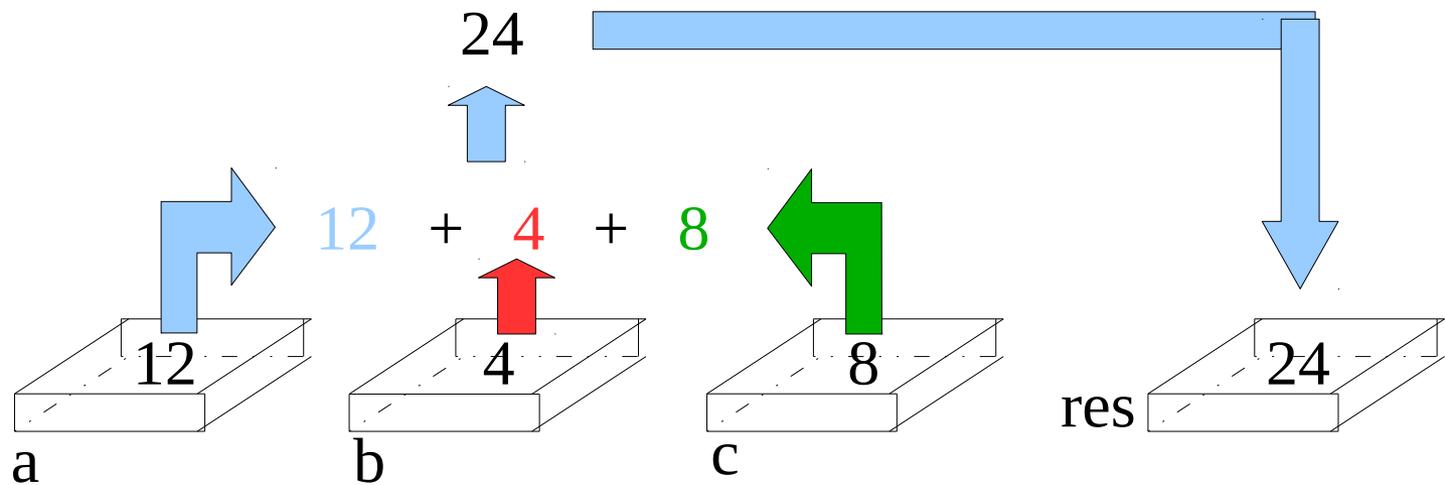
Où stocke-t-on les valeurs dans l'ordinateur ?

Dans des cases avec des noms : a b c

# Exemple moyenne de 3 nombres

Comment fait-on la somme de 3 nombres ?

$a+b+c$  et on met le résultat dans une autre case res



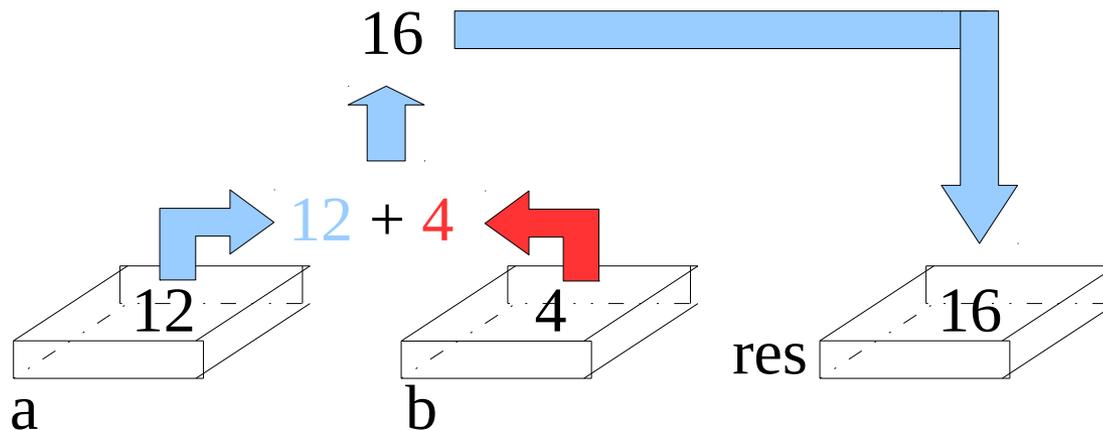
# Exemple

## moyenne de 3 nombres

Comment fait-on la somme de 3 nombres ?

$a+b+c$  et on met le résultat dans une autre case res

OU  $a+b$ , on stocke le résultat dans res,



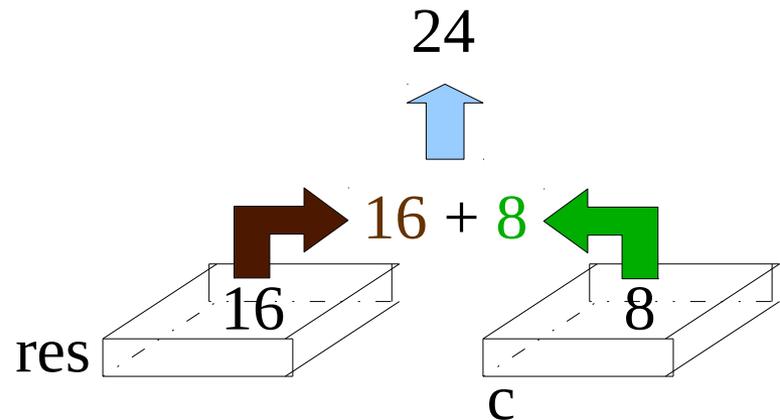
# Exemple

## moyenne de 3 nombres

Comment fait-on la somme de 3 nombres ?

$a+b+c$  et on met le résultat dans une autre case res

OU  $a+b$ , on stocke le résultat dans res, puis on exécute  $res+c$



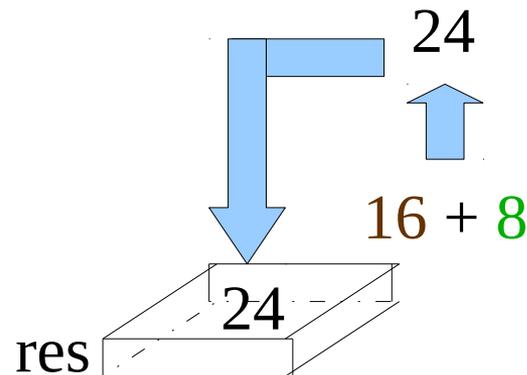
# Exemple

## moyenne de 3 nombres

Comment fait-on la somme de 3 nombres ?

$a+b+c$  et on met le résultat dans une autre case `res`

OU  $a+b$ , on stocke le résultat dans `res`, puis on exécute  $res+c$ , que l'on peut à nouveau mettre dans `res`

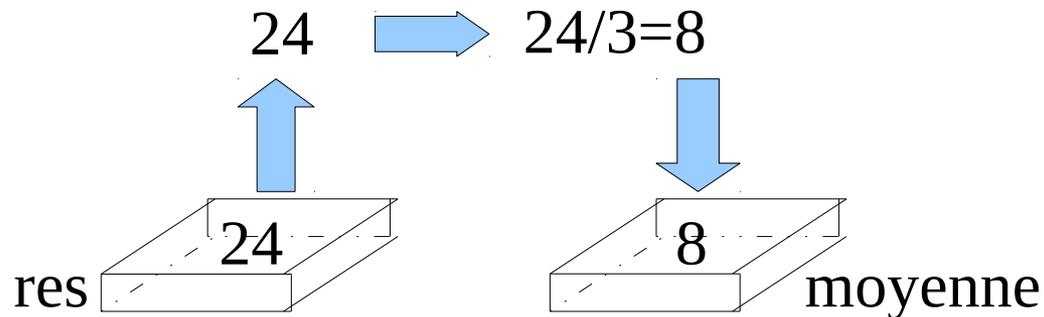


# Exemple

## moyenne de 3 nombres

Comment calcule-t-on la moyenne de 3 nombres ?

On divise la somme contenue dans la variable *res* par 3, et on met le résultat dans une autre variable *moyenne*



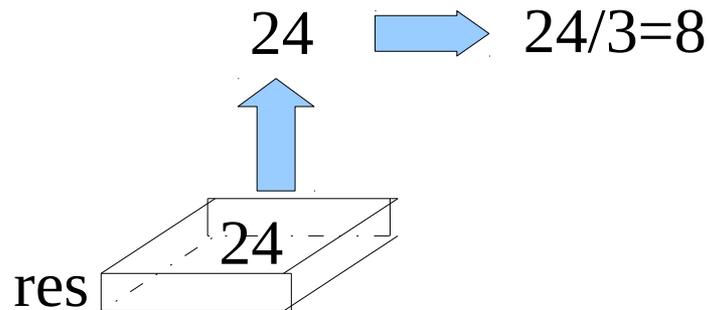
# Exemple

## moyenne de 3 nombres

Comment calcule-t-on la moyenne de 3 nombres ?

On divise la somme contenue dans la variable *res* par 3, et on met le résultat dans une autre variable moyenne

OU On divise la somme contenue dans la variable *res* par 3,



# Exemple

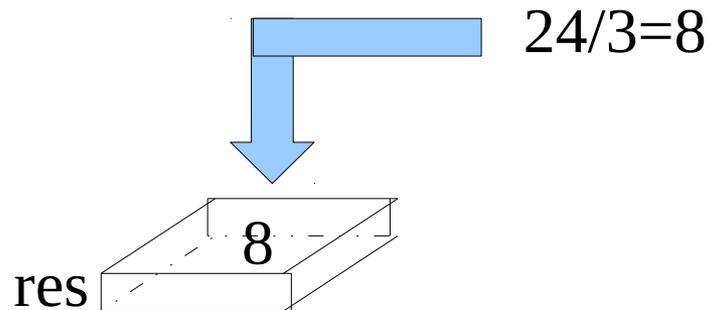
## moyenne de 3 nombres

Comment calcule-t-on la moyenne de 3 nombres ?

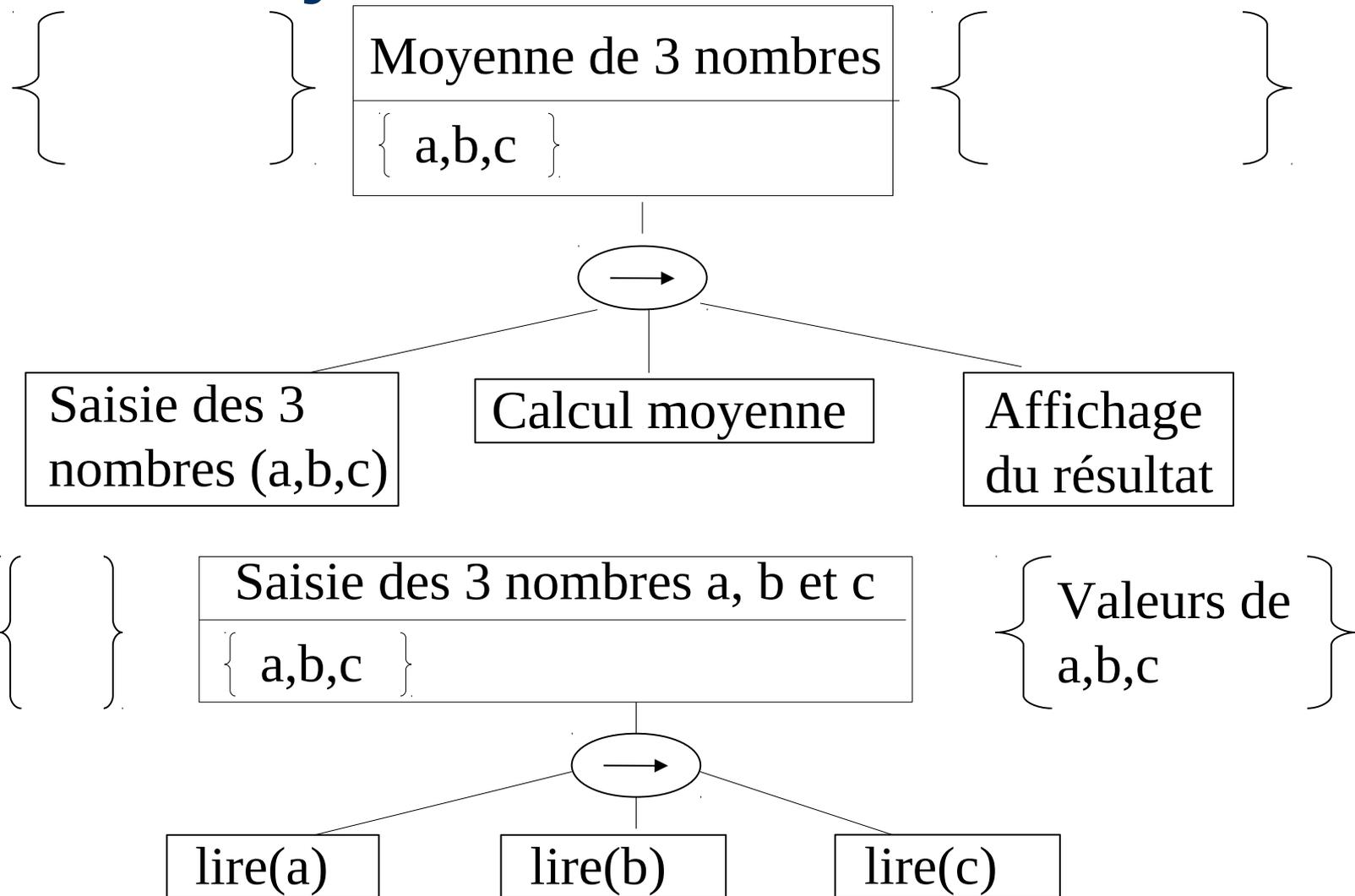
On divise la somme contenue dans la variable *res* par 3, et on met le résultat dans une autre variable moyenne

OU

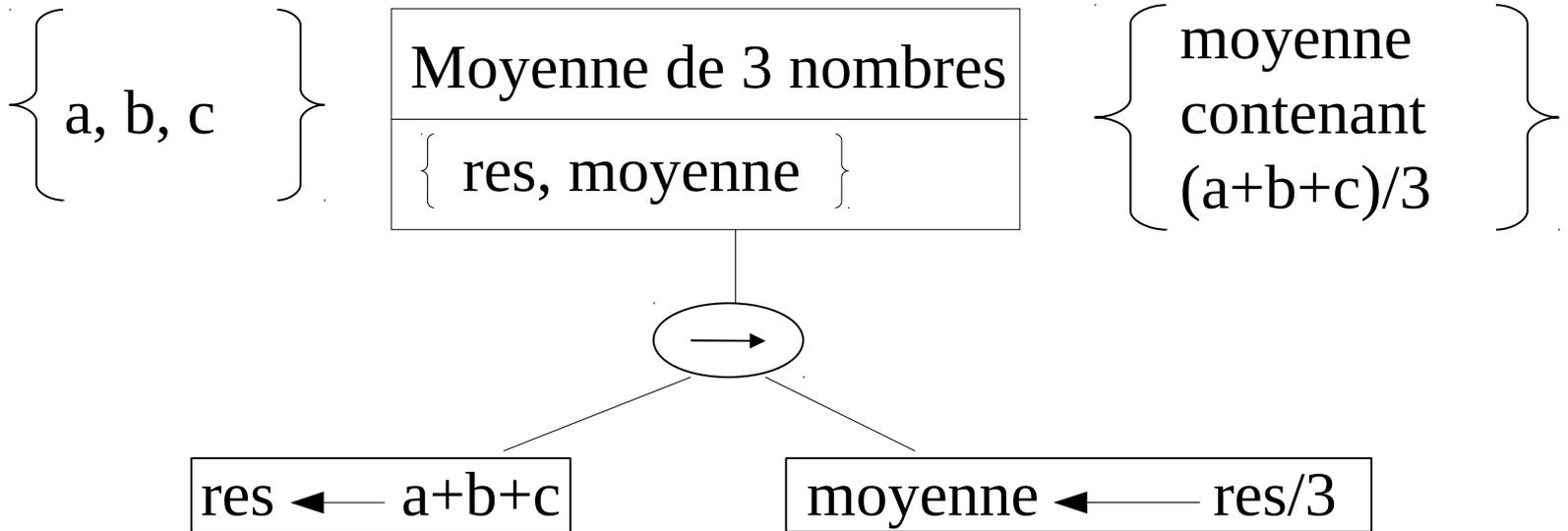
On divise la somme contenue dans la variable *res* par 3, et on peut remettre le résultat dans la même variable *res*



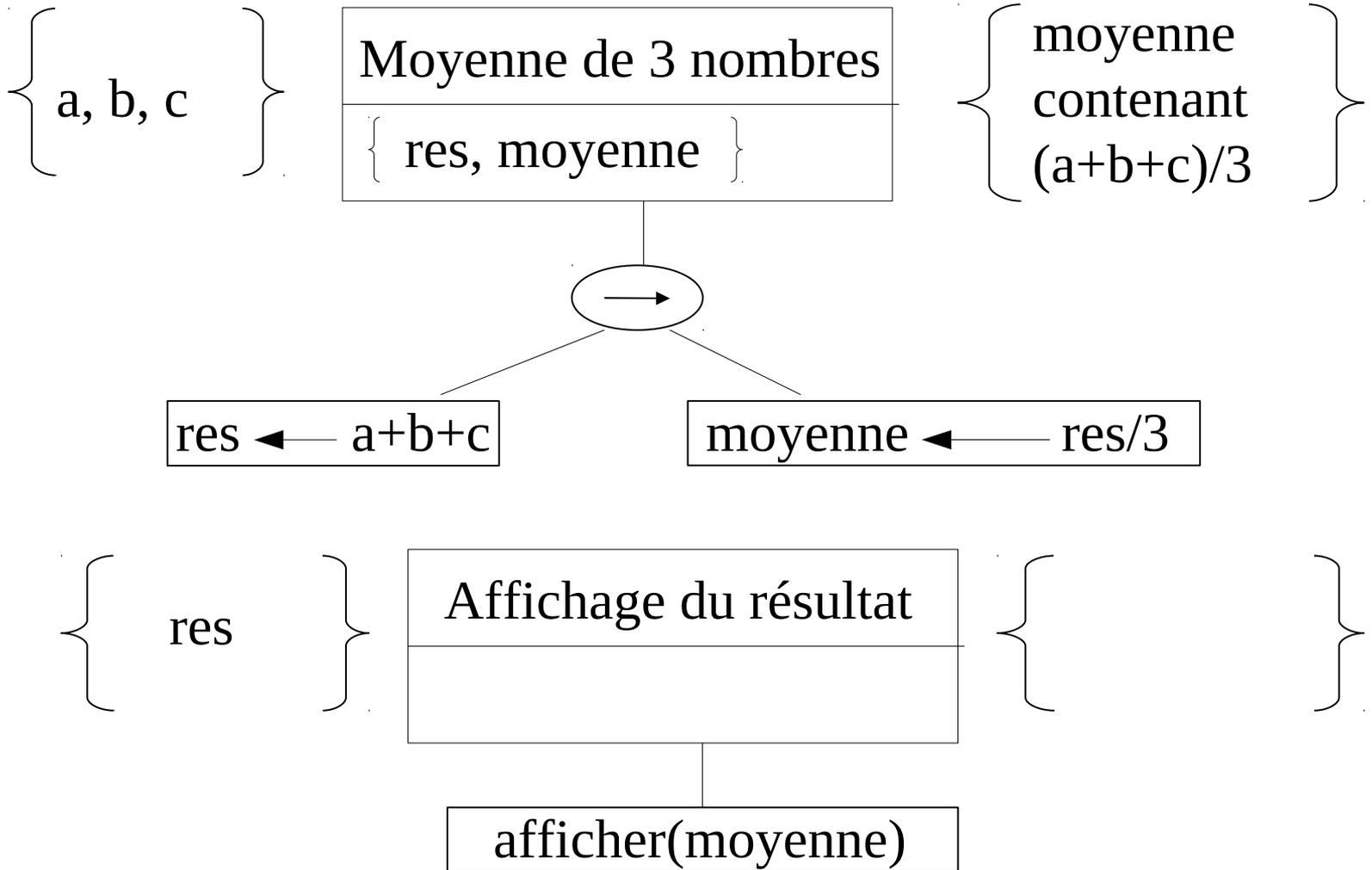
# Algorithme moyenne de 3 nombres

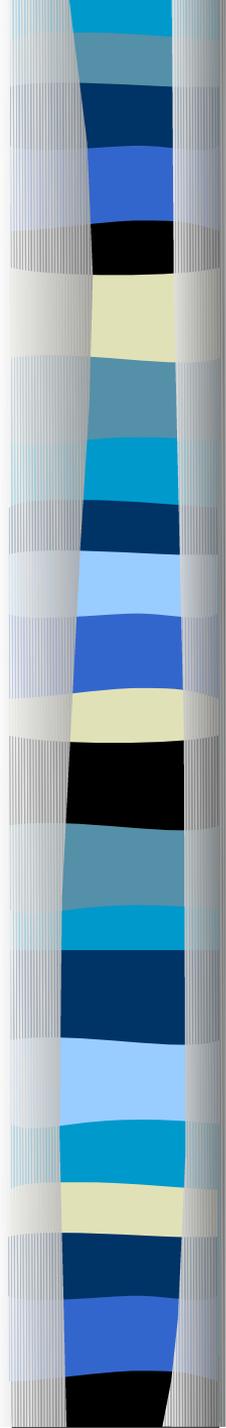


# Algorithme moyenne de 3 nombres



# Algorithme moyenne de 3 nombres





# Exercice

## moyenne de 3 nombres

Réécrire l'algorithme permettant de calculer la moyenne de 3 nombres en :

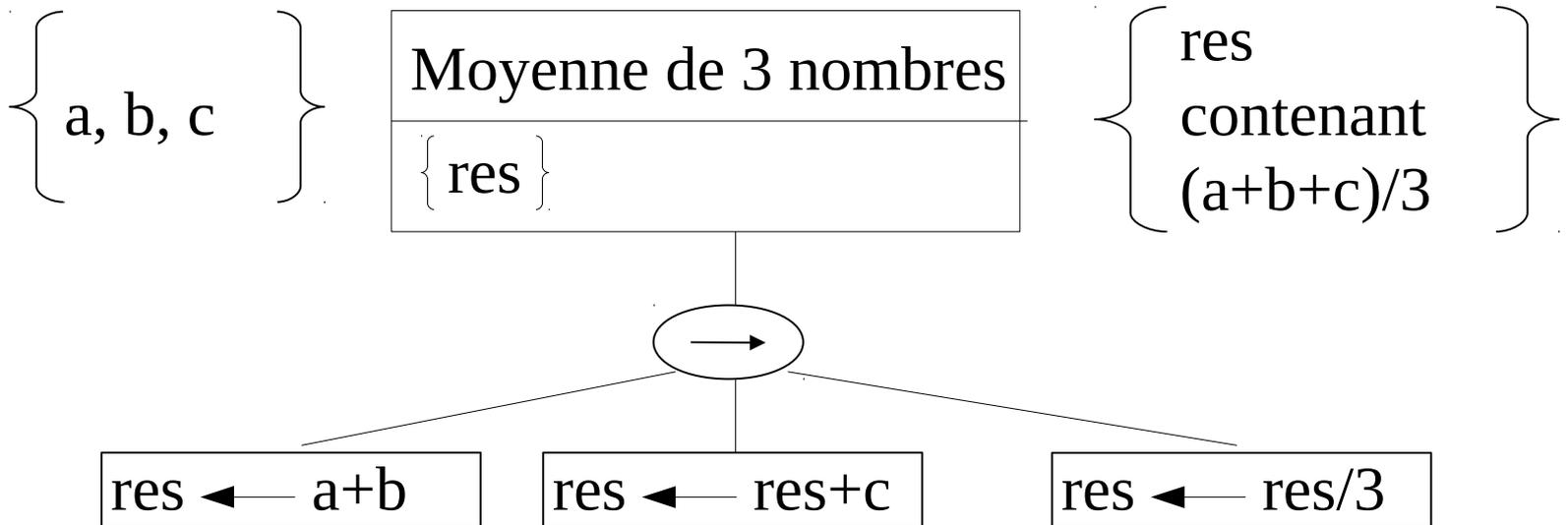
- sommant seulement 2 variables à chaque opération
- utilisant une seule variable pour stocker les calculs

Les 3 phases restent les mêmes (saisie, calcul, affichage), et la phase :

- « saisie » : reste la même
- « calcul » : est modifiée
- « affichage » : au lieu d'afficher *moyenne*, on affichera *res*

# Exercice

## moyenne de 3 nombres





# Les variables

Les variables sont utilisées pour stocker des données.

- Les données peuvent être de différents types :
  - Entiers : -2, 0, 25
  - Réels : -2.0, 12.6, 10.0/3.0
  - Caractères : a, Z, \*

Dans la plupart des langages les variables sont typées

- Les données sont stockées différemment suivant leur type

# Les variables

Lorsque les variables sont typées, certaines opérations peuvent rendre des résultats différents suivant le type des variables.

Exemple : la division

[MCours.com](http://MCours.com)

	théorie	exemple numérique		
	a/b	a	b	a/b
a,b entiers	euclidienne	3	2	1
a,b réels	réelle	3	2	1,5