

Algorithmique : variables, boucles et programmes de calcul

cours et exercices corrigés de maths 3e

« Choisis un nombre, multiplie-le par 4, ajoute 1 » : voilà un programme. Mais c'est aussi une expression littérale, $4x + 1$, et les deux disent rigoureusement la même chose. Tout ce chapitre tient dans ce va-et-vient — exécuter une suite de gestes sur un nombre précis, ou l'écrire une fois pour toutes avec une lettre. La machine fait le premier ; l'algèbre fait le second.

UNE VARIABLE

Une boîte qui retient une valeur, et qu'on peut changer

LE LIEN AVEC L'ALGÈBRE

Un programme de calcul se généralise en une expression

LE PIÈGE

« Ajouter 3 puis multiplier par 4 » donne $4(x + 3)$, pas $4x + 3$

À quoi ça sert : Algorithmique : variables, boucles et programmes de calcul

Un tableur est un algorithme déguisé : chaque cellule contient un programme de calcul, et les autres cellules lui fournissent ses variables. Un jeu vidéo empile des conditions — si la vie tombe à zéro, alors la partie s'arrête — et des boucles qui redessinent l'écran soixante fois par seconde. Mais l'usage le plus proche du collège est le plus banal : une facture d'électricité applique un programme de calcul à votre consommation, un abonnement de bus applique le sien au nombre de trajets. Savoir généraliser un programme en expression, c'est pouvoir vérifier une facture au lieu de la subir.

Un peu d'histoire : Algorithmique : variables, boucles et programmes de calcul

Le mot « algorithme » vient d'un nom propre. Al-Khwârizmî, savant persan de la Maison de la sagesse à Bagdad, écrit vers 820 un traité de méthodes de résolution qui sera traduit en latin sous le titre « Algoritmi de numero indorum ». Le détail remarquable est que le même homme a donné son nom aux deux moitiés de cette fiche : le mot « algèbre » vient du titre d'un autre de ses ouvrages, « al-jabr », qui désigne l'opération consistant à faire passer un terme de l'autre côté d'une équation. Algorithme et algèbre ont donc la même origine, ce qui n'est pas un hasard : ce sont deux façons de décrire une suite d'opérations sans faire le calcul.

Définition : Algorithmique : variables, boucles et programmes de calcul

Un algorithme est une suite d'instructions non ambiguës qui, exécutées dans l'ordre, résolvent un problème. Trois briques suffisent à tout écrire. Une VARIABLE retient une valeur et peut la changer en cours de route. Une BOUCLE répète un bloc d'instructions un nombre donné de fois. Une CONDITION choisit entre deux suites selon qu'un test est vrai ou faux. Un programme de calcul est le cas le plus simple : il n'utilise qu'une variable, et aucune répétition.

Un programme de calcul

On ajoute 3, PUIS on multiplie par 4.

demander Choisis un nombre et attendre

mettre x à réponse

ajouter 3 à x

mettre x à $x \times 4$

dire résultat

Lu dans l'ordre : le nombre choisi reçoit d'abord $+3$, et c'est le TOTAL qui est ensuite multiplié par 4. L'expression est donc $4(x + 3)$ — les parenthèses ne sont pas décoratives, elles disent l'ordre.

Propriétés : Algorithmique : variables, boucles et programmes de calcul

Une variable retient, puis oublie

Une variable est une boîte nommée qui contient une valeur. L'instruction « mettre a à b » copie la valeur de b DANS a : après elle, a vaut ce que valait b , et b n'a pas bougé. La valeur précédente de a , elle, est définitivement perdue. C'est le point qui déroute : le signe ressemble à une égalité, mais il désigne un déplacement à sens unique.

instruction	a	b
au départ	3	7
mettre a à b	7	7
l'ancien 3	perdu	—

b ne change pas

Le contenu va de droite à gauche : b garde sa valeur, a perd la sienne.

Une boucle répète, et le total se construit

Répéter n fois « ajouter p » fait grandir la variable de $n \times p$ au total. Encore faut-il l'avoir INITIALISÉE avant d'entrer dans la boucle : une variable qui n'a pas reçu de valeur de départ ne peut pas être augmentée, et le programme échoue ou repart d'une valeur imprévisible. Le tableau d'exécution — une ligne par tour — est le moyen le plus sûr de suivre ce qui se passe.

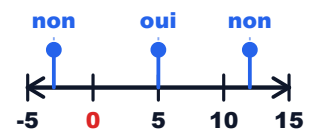
tour	total
au départ	0
après le 1er	5
après le 2e	10
après le 3e	15

3 tours de « ajouter 5 »

Trois répétitions de $+5$ à partir de 0 : le total vaut $3 \times 5 = 15$.

Une condition avec ET : les deux à la fois

La condition « $x > 0$ ET $x < 10$ » n'est vraie que si les DEUX tests le sont en même temps. Un nombre négatif échoue au premier, un nombre trop grand au second, et un seul échec suffit à rendre l'ensemble faux. Avec OU, c'est l'inverse : « $x < 0$ OU $x > 10$ » est vraie dès qu'UN des deux tests réussit.



Trois nombres testés : seul celui du milieu satisfait les deux conditions.

Le contraire d'une condition renverse aussi le bord

Le contraire de « $x \geq 5$ » est « $x < 5$ », et non « $x \leq 5$ ». La valeur 5 elle-même doit changer de camp : elle satisfaisait la première, elle ne doit pas satisfaire la seconde. Oublier ce détail est le bogue le plus courant des programmes qui trient — accepter les notes « supérieures ou égales à 10 » ne s'écrit pas « note > 10 », faute de quoi un élève ayant exactement 10 est refusé.

condition	son contraire
$x \geq 5$	$x < 5$
$x > 5$	$x \leq 5$
ET	devient OU

le bord change de camp

Le symbole strict devient large, et le large devient strict.

Un programme de calcul, et son expression

Généraliser un programme, c'est remplacer le nombre choisi par une lettre et écrire la suite des opérations en une seule expression. « Choisir x , multiplier par 5, enlever 2 » donne $5x - 2$. L'intérêt est immédiat : l'expression vaut pour TOUS les nombres à la fois, alors qu'une exécution ne dit rien de plus que le cas traité.

Multiplier puis enlever

demandeur
Choisis un nombre et attends

mettre x à réponse

mettre x à $x \times 5$

mettre x à $x - 2$

dire résultat

Ce programme calcule $5x - 2$, quel que soit le nombre choisi.

L'ordre décide des parenthèses

C'est le piège central du chapitre, et il vient directement du calcul littéral. « Multiplier par 4 puis ajouter 3 » s'écrit $4x + 3$. Mais « ajouter 3 PUIS multiplier par 4 » s'écrit $4(x + 3)$, car c'est le résultat de l'addition qui est multiplié. Les deux programmes ne donnent pas les mêmes résultats : pour $x = 2$, le premier donne 11 et le second 20.

le programme	l'expression
$\times 4$ puis $+3$	$4x + 3$
$+3$ puis $\times 4$	$4(x + 3)$

l'ordre change tout

Une parenthèse oubliée ne change pas l'apparence : elle change le résultat.

Remonter d'un résultat vers le nombre choisi

Un programme se lit aussi à l'envers. Si $5x + 2$ affiche 17, on cherche le nombre choisi en résolvant $5x + 2 = 17$, ce qui donne $x = 3$. Autre méthode, souvent plus rapide de tête : refaire les opérations inverses dans l'ordre inverse — retrancher 2, puis diviser par 5. Les deux chemins mènent au même nombre, et se contrôlent l'un l'autre.

à l'endroit	à l'envers
$\times 5$	$\div 5$
$+2$	-2
ordre normal	ordre inversé

on défait dans l'autre sens

On commence par défaire la DERNIÈRE opération : ici on enlève 2 avant de diviser.

Généraliser un motif qui grandit

Un motif construit avec des allumettes suit souvent une règle simple : chaque étape ajoute toujours le même nombre. Si le motif 1 en utilise 5, le motif 2 en utilise 8 et le motif 3 en utilise 11, l'écart constant vaut 3 : le motif n en utilise donc $3n + 2$. Le nombre qui s'ajoute à chaque étape devient le coefficient de n , et ce qui reste est ce qu'il fallait au départ.

motif	allumettes	écart
1	5	—
2	8	$+3$
3	11	$+3$
n	$3n + 2$	

l'écart devient le coefficient

Contrôle : pour $n = 1$, $3 \times 1 + 2 = 5$. La formule redonne bien le premier motif.

La formule : Algorithmique : variables, boucles et programmes

de calcul

LE PASSAGE D'UN PROGRAMME DE CALCUL À SON EXPRESSION

« choisir x , ajouter b , multiplier par a » $\longrightarrow$
 $a(x + b)$

⚠ La parenthèse vient de l'ORDRE, pas de la formule. Si l'on multiplie d'abord et qu'on ajoute ensuite, la même paire de nombres donne $ax + b$, qui est une expression différente. Une seule question tranche : sur quoi porte la dernière opération — sur le nombre seul, ou sur tout ce qui précède ?

l'ordre des gestes	l'expression
+ b puis × a	$a(x + b)$
× a puis + b	$ax + b$
× a puis × a	a^2x

la dernière opération porte sur tout

Deux programmes équivalents donnent même résultat pour TO — pas seulement pour

Méthode : Algorithmique : variables, boucles et programmes de calcul

1. Exécuter un programme : le tableau d'exécution

Une ligne par étape, une colonne par variable. On y inscrit la valeur APRÈS chaque instruction. C'est long à écrire et cela ne se trompe jamais, alors que suivre un programme de tête échoue dès qu'une boucle dépasse trois tours.

2. Généraliser : remplacer le nombre par une lettre

On réécrit la suite des gestes en gardant x tout du long, sans jamais calculer. À chaque étape, on se demande sur quoi porte l'opération : sur x seul, ou sur tout ce qu'on a déjà obtenu ? La réponse décide des parenthèses.

3. Vérifier une généralisation sur un nombre

On exécute le programme sur une valeur simple, puis on calcule l'expression pour cette même valeur. Les deux doivent coïncider. ⚠ Un seul essai ne prouve pas l'équivalence, mais un désaccord prouve l'erreur — et c'est ce qu'on cherche.

4. Chercher un bogue : les quatre suspects

Une variable non initialisée avant une boucle ; un test strict là où il fallait un test large ; un affichage placé avant le calcul ; et l'ordre des opérations qui inverse les parenthèses. La banque du coach ne pose pratiquement que ces quatre-là.

5. Écrire une condition composée

ET exige les deux tests, OU se contente d'un seul. Pour être sûr, on teste trois nombres : un trop petit, un dans la cible, un trop grand. Si les trois réponses sont celles attendues, la condition est juste.

Selon ce que l'on cherche : Algorithmique : variables, boucles et programmes de calcul

On me donne un programme et un nombre

J'exécute, étape par étape, dans un tableau d'exécution. Une ligne par instruction, la valeur après chacune.

On me demande l'expression du programme

Je garde x et je n'effectue aucun calcul. Je surveille l'ordre : la dernière opération porte sur tout ce qui précède.

On me donne le résultat et on cherche le nombre choisi

Je résous l'équation, ou je défais les opérations dans l'ordre inverse — les deux se contrôlent.

On me demande si deux programmes sont équivalents

J'écris les deux expressions et je les développe. Si elles se réduisent à la même, les programmes sont équivalents pour tout x .

Le programme contient une boucle et une condition

J'exécute d'abord toute la boucle dans un tableau, et je teste la condition SEULEMENT sur la valeur finale.

Exemples corrigés : Algorithmique : variables, boucles et

programmes de calcul

Une boucle, puis un test

La variable score vaut 0. On répète 4 fois : ajouter 7 à score. Puis, si $\text{score} \geq 30$, on affiche « réussi », sinon « raté ».

Quel message s'affiche ?

tour	score
départ	0
1	7
2	14
3	21
4	28

on teste la valeur finale

28 est inférieur à 30 : le message est « raté ».

On exécute d'abord la boucle en entier : partant de 0, quatre ajouts de 7 donnent 7, puis 14, puis 21, puis 28. La valeur finale est 28 — on peut aussi l'obtenir directement par $4 \times 7 = 28$. On teste ENSUITE, et une seule fois : $28 \geq 30$ est faux. Le message affiché est donc « raté ». ⚠ L'erreur consiste à tester à chaque tour ; ici la condition vient après la boucle, pas dedans.

Généraliser un programme

On choisit un nombre, on lui ajoute 3, puis on multiplie le résultat par 2.

Quelle expression généralise ce programme ?

On appelle x le nombre choisi. Après l'addition, on a $x + 3$. La multiplication porte ensuite sur CE résultat, donc sur la somme entière : il faut des parenthèses, et l'expression est $2(x + 3)$. Écrire $2x + 3$ décrirait un autre programme, celui où l'on multiplie d'abord. Contrôle sur un nombre : pour $x = 5$, le programme donne $5 + 3 = 8$ puis $8 \times 2 = 16$; et $2(5 + 3) = 2 \times 8 = 16$. C'est cohérent, alors que $2 \times 5 + 3$ donnerait 13.

Corriger un programme

Un programme doit ajouter 3 à x puis multiplier par 4. Il calcule $4x + 3$.

Où est l'erreur, et comment la corriger ?

L'expression écrite correspond à l'ordre inverse : elle multiplie d'abord, puis ajoute. Or l'énoncé demande que la multiplication porte sur le résultat de l'addition. Il manque donc des parenthèses, et la bonne expression est $4(x + 3)$. On peut le vérifier sur un nombre : pour $x = 2$, le programme voulu donne $2 + 3 = 5$ puis $5 \times 4 = 20$, tandis que $4 \times 2 + 3$ donne 11. L'écart de 9 confirme que ce n'était pas un détail d'écriture.

Le motif d'allumettes

Un motif utilise 5 allumettes à l'étape 1, 8 à l'étape 2, et 11 à l'étape 3.

Combien en faut-il pour l'étape 20 ?

étape	allumettes
1	5
2	8
3	11
20	62

$$3n + 2$$

La formule évite de dessiner les vingt motifs.

On cherche d'abord l'écart entre deux étapes consécutives : $8 - 5 = 3$, et $11 - 8 = 3$. L'écart est constant et vaut 3, qui devient donc le coefficient de n . Reste à ajuster : $3 \times 1 = 3$, alors que l'étape 1 en demande 5 — il manque 2. La formule est $3n + 2$. On la contrôle sur une autre étape que celle qui a servi : $3 \times 3 + 2 = 11$, ce qui est bien le motif 3. Pour l'étape 20 : $3 \times 20 + 2 = 62$ allumettes.

Pièges à éviter : Algorithmique : variables, boucles et programmes de calcul

Écrire $4x + 3$ pour « ajouter 3 puis multiplier par 4 ». La multiplication porte sur la somme : c'est $4(x + 3)$.

Oublier d'initialiser une variable avant une boucle qui l'augmente : elle n'a alors aucune valeur de départ, et le total obtenu ne veut rien dire.

Tester « note > 10 » pour accepter les notes supérieures OU ÉGALES à 10 : l'élève ayant exactement 10 est refusé. Il faut $\geq$.

Croire que « mettre a à b » échange les deux variables. La valeur va de b vers a ; b ne change pas, et l'ancienne valeur de a est perdue.

Confondre ET et OU. Avec ET, un seul test raté suffit à tout invalider ; avec OU, un seul test réussi suffit à tout valider.

Placer l'affichage du résultat avant le calcul : le programme montre alors une valeur qui n'a pas encore été mise à jour.

Vérifier une généralisation sur le nombre qui a servi à la trouver : cela ne prouve rien. Il faut en essayer un autre.

À retenir : Algorithmique : variables, boucles et programmes de calcul

Un programme de calcul se généralise en une expression littérale.

L'ordre des gestes décide des parenthèses : $a(x + b)$ n'est pas $ax + b$.

« Mettre a à b » copie b dans a : b ne bouge pas, l'ancien a est perdu.

Une variable augmentée dans une boucle doit être initialisée avant.

ET exige les deux conditions ; OU se contente d'une seule.

Le contraire de $x \geq 5$ est $x < 5$: le bord change de camp.

Deux programmes sont équivalents s'ils donnent le même résultat pour TOUT x .

Exercices corrigés : Algorithmique : variables, boucles et programmes de calcul

1. La variable a vaut 3 et b vaut 7. On exécute « mettre a à b ». Que valent a et b ensuite ?

Correction :

a vaut 7 et b vaut toujours 7. La valeur est copiée de b vers a ; l'ancienne valeur de a , qui était 3, est perdue.

2. La variable total vaut 0. On répète 6 fois : ajouter 4 à total. Quelle est la valeur finale ?

Correction :

Six ajouts de 4 à partir de 0 donnent $6 \times 4 = 24$. On peut le vérifier en déroulant : 4, 8, 12, 16, 20, 24.

3. La variable x vaut 12. La condition « $x > 0$ ET $x < 10$ » est-elle vraie ?

Correction :

Non. Le premier test réussit — 12 est bien positif — mais le second échoue, car 12 n'est pas inférieur à 10. Avec ET, un seul échec suffit à rendre l'ensemble faux.

4. Quel est le contraire de la condition « $x \geq 5$ » ?

Correction :

C'est « $x < 5$ ». Le symbole large devient strict, de sorte que la valeur 5 change de camp : elle satisfaisait la première condition, elle ne satisfait pas la seconde.

5. On choisit un nombre, on le multiplie par 5, puis on enlève 2. Quelle expression correspond ?

Correction :

$5x - 2$. La multiplication vient en premier et porte sur le nombre seul : aucune parenthèse n'est nécessaire.

6. On choisit un nombre, on lui ajoute 4, puis on multiplie par 3. Quelle expression correspond ?

Correction :

$3(x + 4)$. La multiplication porte cette fois sur le résultat de l'addition, donc les parenthèses sont obligatoires. Pour $x = 1$: le programme donne 15, et $3(1 + 4) = 15$.

7. Un programme calcule $3x + 4$. Le résultat affiché est 19. Quel nombre avait-on choisi ?

Correction :

On résout $3x + 4 = 19$, donc $3x = 15$, donc $x = 5$. Par les opérations inverses : on enlève 4, ce qui donne 15, puis on divise par 3, ce qui donne 5.

8. Un programme doit additionner des nombres, mais oublie d'initialiser le total. Quel est le problème ?

Correction :

Le total n'a pas de valeur de départ : la première addition n'a rien à quoi s'ajouter. Le résultat est soit une erreur, soit une valeur laissée par une exécution précédente — donc imprévisible.

9. Un motif utilise $2n + 5$ allumettes à l'étape n . Combien en faut-il à l'étape 12 ?

Correction :

On remplace n par 12 : $2 \times 12 + 5 = 24 + 5 = 29$ allumettes.

10. Deux programmes de calcul donnent le même résultat pour toutes les valeurs de x . Comment dit-on ?

Correction :

On dit qu'ils sont ÉQUIVALENTS. Cela se prouve en développant les deux expressions : si elles se réduisent à la même, l'égalité vaut pour tout x — un seul nombre commun ne suffirait pas.