

Algorithmique (1) : Rappels

Définition :

Un algorithme est une suite finie d'instructions données dans un certain ordre, qui une fois exécutée, permet d'obtenir un résultat.

Le programme sert ensuite à automatiser un calcul.


Structure d'un algorithme

Un algorithme peut être décomposé en **trois parties** :

1. On entre des données (ou variables) C'est la phase d' ENTREE
2. Ces données sont traitées par une suite d'instructions C'est la phase de TRAITEMENT
3. Un résultat est obtenu C'est la phase de SORTIE

Présentation d'un algorithme en pseudo-langage et instructions de base

Voici un algorithme écrit en pseudo-langage.
On donne deux présentations (les plus courantes)

<p>Variables x, y et z sont des nombres réels</p> <p>Début de l'algorithme Saisir x y prend la valeur $(x-3)^2$ z prend la valeur $-2y+5$ Afficher z</p> <p>Fin de l'algorithme</p>	<p>Variables x, y et z sont des nombres réels</p> <p>Entrée Saisir x</p> <p>Traitement y prend la valeur $(x-3)^2$ z prend la valeur $-2y+5$</p> <p>Sortie Afficher z</p>	<p>Dans certains algorithmes, on pourra ajouter une phase « Initialisation »</p> 
--	---	--

Instructions de base d'un algorithme

INSTRUCTIONS DE BASE D'UN ALGORITHME ENTREE-TRAITEMENT SORTIE Pseudo-langage	Commentaires
	<i>Cette écriture n'est pas « figée », elle peut varier selon les manuels utilisés. C'est une écriture intermédiaire entre le langage courant et les différents langages de programmation sur ordinateur ou calculatrice.</i>
Saisir x	<p>Entrée : Saisie ou initialisation des variables</p> <p>On peut se représenter une variable comme une boîte portant une étiquette ("le nom de la variable") à l'intérieur de laquelle on peut placer un contenu, une valeur.</p> <p>L'instruction de base « Saisir x » signifie trois choses :</p> <ol style="list-style-type: none"> 1) elle permet de demander à l'utilisateur d'entrer une valeur 2) elle définit le nom de la variable où va être rangée la réponse de l'utilisateur 3) elle stocke la valeur entrée par l'utilisateur dans la variable x. <p><i>Dans certains manuels, on utilise plutôt l'instruction « Lire x » ou « Entrer x »</i></p>
y prend la valeur $(x-3)^2$	<p>Traitement : Opération sur les variables</p> <p>Cette instruction de base est appelée affectation :</p> <p>Elle permet d'attribuer (affecter) une valeur à une variable.</p> <p><i>Ici on attribue la valeur $(x-3)^2$ à la variable y.</i></p>
z prend la valeur $-2y+5$	<p style="text-align: center;">« Affecter revient à remplir la boîte (la variable !) »</p> <p><i>Dans certains manuels, on utilise plutôt l'instruction</i></p> <p style="text-align: center;">« Affecter à y la valeur $(x-3)^2$ » Ou encore « mettre $(x-3)^2$ dans y »</p>

Afficher z	<p align="center">Sortie : Affichage des résultats</p> <p>Cette instruction de base permet d'afficher la valeur contenue dans la variable z. (attention, ce n'est pas la lettre z qui est annoncée) <i>On rencontrera aussi l'expression « Sortir z » qui aura le même sens.</i></p>
------------	---

Exemple :

Testons l'algorithme en choisissant $x = 7$. Alors :

$$y = (7 - 3)^2 = 16$$

$$z = -2 \times 16 + 5 = -27$$

L'algorithme affiche à la sortie le nombre -27.

Remarque : Il ne faut pas confondre l'affichage d'un message et l'affichage de la valeur d'une variable :

- **Afficher z** affiche sur le logiciel la valeur de la variable z (mais n'affiche pas la lettre z !)
- **Afficher "z"** affichera la lettre « z » sur l'écran.

Récapitulatif des instructions utilisées en programmation :

	Casio	Texas	AlgoBox
Créer un nouveau programme	Menu PRGM (touches SHIFT VARS), puis NEW (touche F3)	prgm , puis NOUV	Cliquer sur Nouveau
Entrer A	? → A	Input A	+ Ajouter LIRE Variable
Entrer A, en affichant « A = ? »	« A = » ? → A	Prompt A	
Afficher A	A	Disp A	+ Ajouter AFFICHER Variable
Afficher un texte, par exemple « oui »	« oui »	Disp « oui »	+ Ajouter AFFICHER Message
Opérateurs relationnels (comme =, ≤, ≥, ≠)	PRGM (touches SHIFT VARS), puis , puis REL	tests (touches 2nde math)	Au clavier
Affecter à B la valeur de A	Taper A puis la touche et enfin B	Taper A puis la touche STO→ et enfin B	+ AFFECTER valeur à variable
Pour I variant de I ₀ à N Faire {instructions} Fin Pour	For I ₀ → I to N {instructions} Next (dans PRGM , puis COM)	For (I, I ₀ , N) {instructions} End (dans prgm , puis CTL)	+ Ajouter POUR... DE...A
Si {condition C} Alors {instructions A} Sinon {instructions B} Fin Si	If {condition C} Then {instructions A} Else {instructions B} IfEnd (dans PRGM , puis COM)	If {condition C} Then {instructions A} Else {instructions B} End (dans prgm , puis CTL)	+ Ajouter SI...ALORS
Tant que {condition C} Faire {instructions} Fin Tant que	While {condition C} {instructions} WhileEnd (dans PRGM , puis COM)	While {condition C} {instructions} End (dans prgm , puis CTL)	+ Ajouter TANT QUE...