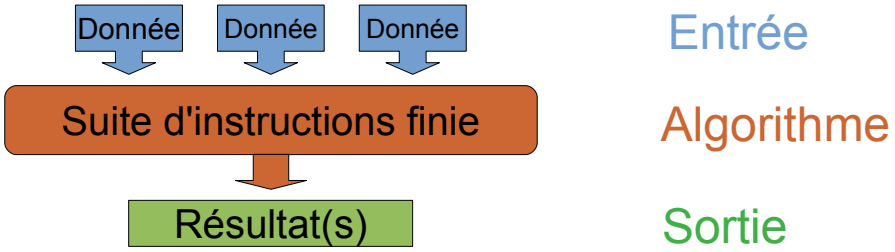
Algorithmique 1

Qu'est-ce-qu'un algorithme ?

Un **algorithme** est une suite finie d'instructions qui permettent de résoudre un certain problème. Il permet, à partir d'informations initiales, de produire de nouvelles informations (une réponse à une question, de nouvelles quantités numériques, etc.).

Le mot *algorithme* vient du nom du mathématicien Al Khuwarizmi qui, au IXe siècle, écrivit un livre important sur la résolution des équations (le mot *algèbre* vient du titre de ce livre).

Retenez ce schéma :



Un algorithme peut être décrit :

- en **langage naturel** ;
- en **pseudo-code** ;
- dans un **langage informatique** (on parle alors de **programme**).

Exercice 1 : algorithme en langage naturel

Voici un algorithme :

- choisir un nombre ;
- lui ajouter 2 et multiplier le tout par 3 ;
- mettre au carré le résultat ;
- afficher le résultat.

1°) Complétez le tableau suivant :

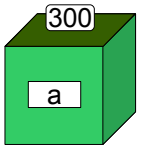
Nombre choisi	0	1	4	5
Résultat de l'algorithme				

2°) Soit x le nombre choisi au départ. Donnez l'expression du résultat affiché par l'algorithme en fonction de x.

3°) Retrouvez les résultats du 1°) en utilisant le menu « Fonction » de votre calculatrice et l'option « Afficher les valeurs » (ou « Table » suivant les machines).

Les variables informatiques

En programmation, il faut souvent stocker des données pour les utiliser plus tard. Les **variables** sont faîtes pour cela. Supposons que je veuille stocker la valeur 300, je peux utiliser une variable nommée par exemple « a », cette variable peut être vue comme une boîte :



En langage naturel, je peux écrire : a prend la valeur 300

Le nom des variables est leur *identificateur*, il permet de les utiliser plus simplement.

L'action consistant à mettre une valeur dans une variable (exemple : f ← 300) est une **affectation**.

Ici, par exemple, nous avons affecté à la variable a la valeur 300.

Exercice II : quelques affectations

Donnez la valeur contenue dans la variable *d* à la fin de l'algorithme suivant :

Algorithme
a prend la valeur 5
b prend la valeur – 1
c prend la valeur de *a* + b × 2
d prend la valeur de *c* + 4

Pseudo-code

Le pseudo-code est une façon d'écrire des algorithmes, qui est entre le langage naturel et les langages de programmation.

Par exemple, l'instruction	a prend la valeur 300	
peut s'écrire, en pseudo-code :	300 → a	(on met 300 dans la boîte « a »)
ou comme cela :	a ← 300	(plus fréquent)

Exercice III : traduction en pseudo-code

Traduisez l'algorithme de l'exercice II en pseudo-code :

Algorithme en langage naturel	**Algorithme en pseudo-code**
a prend la valeur 5	
b prend la valeur – 1	
c prend la valeur de *a* + b × 2	
d prend la valeur de *c* + 4	

Exercice IV : traduction en pseudo-code

Voici un exemple de traduction de l'algorithme de l'exercice I :

Algorithme en langage naturel	Algorithme en pseudo-code
• étant donné un nombre ; • lui ajouter 2 et multiplier le tout par 3 ; • mettre au carré le résultat ; • afficher le résultat.	y ← x + 2 z ← y*3 a ← z^2 Afficher a

1°) Traduire de même l'algorithme suivant en pseudo-code :

- étant donnés deux nombres entiers ;
- multiplier le premier nombre par 2 ;
- multiplier le second nombre par 3 ;
- ajouter les deux résultats précédents ;
- afficher le dernier résultat.

2°) Écrire une autre version de cet algorithme utilisant le moins de variables possible.

Effets d'une affectation

Effet 1 : écrasement

Si j'exécute à la suite les deux instructions :

a ← 2

a ← 3

alors la variable « a » contiendra la valeur 3 et pas 5.

Le stockage d'une valeur dans une variable **écrase (efface) le contenu précédent**.

Si je veux placer 2 dans la variable « a » puis lui ajouter 3, je dois faire ainsi :

a ← 2
a ← a + 3

Effet 2 : copie

A la suite des trois instructions :

a ← 2
b ← 7
b ← a

les deux variables « a » et « b » contiendront 2 (le contenu de a est mis dans a à la fin).
L'instruction « b ← a » copie la valeur de « a » pour la mettre dans « b ». Elle ne vide pas le contenu de « a » (contrairement à une boîte qu'on viderait).

Exercice V

1°) Pour chacune des suites d'instructions suivantes, créez un tableau de la forme ci-dessous au fur et à mesure du déroulement de l'algorithme pour connaître les valeurs finales des variables :

Étapes \ Variables	**a**	**b**
a ← 2	2	//////////
.....	.....	.....

a)
```
a ← 2
b ← 3
a ← b + 5
b ← a – 2
```

b)
```
a ← 5
b ← 2
c ← a – 3
a ← b + 4
b ← c – 1
```

c)
```
a ← 1
b ← a + 3
a ← a – 1
b ← a + 1
```

2°) a) Quelles sont les valeurs de a et b à la fin de cet algorithme ?

```
a ← 5
b ← 1
a ← b
b ← a
```

b) Les deux dernières affectations échangent-elles les valeurs de a et de b ?
c) Modifiez cet algorithme afin qu'il échange les valeurs de a et de b.

Remarque importante : une affectation n'est pas une égalité
L'instruction « b ← a » s'écrit souvent « b = a » en langage informatique. Cependant ceci n'est pas une égalité au sens mathématique mais une *affectation*. Ainsi :

- « b ← a » ne revient pas à « a ← b » donc « b = a » ne revient pas à « a = b » !!
- l'affectation « a = a + 1 », très fréquente en informatique, serait fausse en tant qu'égalité !
- en informatique, l'instruction « 300 = a » ne veut rien dire (on ne peut pas mettre a dans 300 !).

Exercice VI : pour ceux qui ont fini (d'après Wikipédia)

L'algorithme suivant a été trouvé par le mathématicien indien Dattatreya Ramachandra Kaprekar. Il consiste à associer à un nombre entier n un autre nombre $K(n)$ généré de la façon suivante :

- on forme le nombre n_1 en arrangeant les chiffres du nombre n dans l'ordre croissant et le nombre n_2 en les arrangeant dans l'ordre décroissant ;
- on pose $K(n) = n_2 - n_1$;
- on recommence ensuite le processus avec $K(n)$ autant de fois que nécessaire...

1°) a) On choisit $n = 59$. Vérifiez que le calcul indiqué ci-dessus donne 36.

b) Continuez d'appliquer le processus à partir de 36. Que remarque-t-on ?

2°) Choisissez maintenant le numéro de votre jour de naissance (par exemple, 23 si vous êtes né le 23 juillet 2007) et appliquez lui l'algorithme. Que remarquez-vous ?

3°) Choisissez maintenant le nombre formé du jour et du mois de votre date de naissance (par exemple, 2307 si vous êtes né le 23 juillet 2007) et appliquez-lui l'algorithme. Que remarquez-vous ?