Notion d'algorithmique

I/ Des exemples d'algorithmes au quotidien

Sans le savoir, on utilise quotidiennement des algorithmes comme la recette de cuisine ci-dessous.

EXEMPLE 1 : UNE RECETTE DE CUISINE

Voici une recette tirée d'un livre de cuisine :
Se procurer 250 g de chocolat noir, 250 g de beurre, 4 œufs, 250 g de sucre et 75 g de farine.

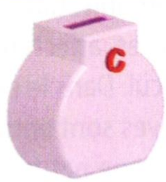
a. Faire fondre le chocolat au bain-marie ; ajouter le beurre, mélanger ; ajouter la farine.
b. Battre les œufs en omelette ; ajouter le sucre et mélanger.
c. Mélanger les deux préparations.
d. Verser dans un moule et faire cuire 45 minutes au four à 220 °C.
Laisser refroidir, puis servir le gâteau.

Ce texte décrit les opérations à réaliser successivement pour faire un moelleux au chocolat.

A partir des ingrédients de la recette, avec les quantités requises, le texte donne les **règles** à suivre : il s'agit des **étapes** a, b, c et d qui **s'enchaînent**. Le **résultat** est le gâteau fini.

EXEMPLE 2 : LA TIRELIRE DE CORALIE

Pour ses 7 ans, Coralie reçoit une tirelire contenant 10 €.
Pour ses 8 ans, elle reçoit 20 € qu'elle met dans la tirelire.
Pour ses 9 ans, elle reçoit 30 € qu'elle met dans la tirelire.
Elle casse alors sa tirelire et récupère l'argent.



Au départ, Coralie a une tirelire garnie. Après **deux étapes** au cours desquelles le **contenu** de la tirelire **se modifie**, elle découvre le **résultat** en cassant sa tirelire.

EXEMPLE 3 : UN PROGRAMME DE CALCUL

Choisir un nombre de départ
Multiplier ce nombre par −2
Ajouter 5 au produit
Multiplier le résultat par 3
Écrire le résultat obtenu

Ce programme de calcul demande d'abord de choisir un nombre.
Si on choisit 10, on calcule d'abord : $10 \times (-2) = -20$. Puis on ajoute 5, ce qui donne −15. On multiplie le nombre par 3, ce qui donne −45.
L'application de la suite des règles données a conduit au résultat que l'on écrit : −45.

Algorithme : ....

....

A noter : Le mot « algorithme » vient du nom mathématicien persan **Al-Khwarizmi** (780 – 850). il a écrit en langue arabe le plus ancien traité d'algèbre dans lequel il décrivait des procédures de résolution pas à pas d'équations.

II/ Ecriture d'un algorithme

1/ Structure d'un algorithme

Un algorithme comprend :

- une phase **d'initialisation** : on ***déclare*** et ***initialise*** les variables et on ***entre*** les données ;
- une phase de **traitement** du problème ;
- une phase de **sortie** des résultats.

Exemple :

Soit deux nombres réels a et b.
(1) Calculer $(a+b)^2$ et affecter à c le résultat.
(2) Calculer $(a-b)^2$ et affecter à d le résultat.
(3) Calculer $\frac{c-d}{4}$ et affecter à s le résultat.
Donner le résultat obtenu.

- La phase d'initialisation est le choix de a et b : on dit qu'on entre les données a et b. On écrira : **Saisir *a*, *b*.**
- La phase de traitement est formée des trois calculs successifs décrits en **(1)**, **(2)**, **(3)**.
- La phase de sortie permet de donner le nombre obtenu après cette suite de calculs. On écrira : **Afficher *s*.**

On a utilisé dans cet algorithme les variables a, b, c, d et s.

2/ Formalisme d'écriture d'un algorithme

Ecriture d'un algorithme en langage naturel

Soit deux nombres réels a et b.
(1) Calculer $(a+b)^2$ et affecter à c le résultat.
(2) Calculer $(a-b)^2$ et affecter à d le résultat.
(3) Calculer $\frac{c-d}{4}$ et affecter à s le résultat.
Donner le résultat obtenu.

En langage formel

Variables	a, b, c, d, s sont des réels
Entrées	Saisir a et b
Traitement	Affecter à c la valeur $(a+b)^2$
	Affecter à d la valeur $(a-b)^2$
	Affecter à s la valeur $(c-d)/4$
Sortie	Afficher s

Exemple :

Voici un algorithme.

(1) x, y et z sont des nombres réels
(2) Saisir x et y
(3) z prend la valeur $x+y$
(4) x prend la valeur x^2+1
(5) y prend la valeur z/x
(6) Afficher y

1/ Quelles sont les **variables** ?
....

2/ **Déterminer** dans l'algorithme la (ou les) ligne(s) qui correspondent aux étapes :
les entrées : .... ;
le traitement : .... ;
la sortie : ....

Exercice 1 :

Voici un algorithme :

Variables	u, x et y sont des entiers
Entrée	Saisir x
Traitement	u prend la valeur $x + 4$
	y prend la valeur $u \times x$
Sortie	Afficher y

Donner la valeur obtenue en sortie pour $x = 3$, puis pour $x = -1$.

....

....

Exercice 2 :

Voici un algorithme écrit en langage naturel.

Choisir un nombre
Lui ajouter 1
Doubler le résultat précédent
Enlever 3 au résultat
Donner le résultat obtenu

Réécrire l'algorithme précédent en utilisant la structure ci-dessous.

Variables	x est ....
Entrée	Saisir ....
Traitement	x prend la valeur ....
	x prend la valeur ....
	....
Sortie	Afficher ....

Exercice 3 :

Ecrire un algorithme affichant la somme S et le produit P de deux entiers a et b donnés.

....

....

....

....

....

....

....

....

....

....

Exercice 4 :

Ecrire un algorithme qui demande une température C (exprimée en degrés Celsius), puis la transforme en degrés Fahrenheit F, sachant que l'on a la relation F = 1,8 x C + 32.

....

....

....

....

....

....

....

....

III/ Instruction conditionnelle

Une **instruction conditionnelle** permet d'exécuter **une partie** d'un algorithme en fonction d'une **condition** (vérifiée ou non) fixée par le programmeur.

Voici la structure :

```
Si {condition C}
 | Alors {instructions A}
 | Sinon {instructions B}
Fin Si
```

Si la condition C est vérifiée, seules les instructions A sont exécutées.
Si la condition C n'est pas vérifiée, seules les instructions B sont exécutées.

Remarque : on peut aussi utiliser la structure incomplète : « Si … Alors … » : dans ce cas, si la condition C n'est pas vérifiée, l'exécution de l'algorithme continue après le Fin Si.

Exemple :

On considère l'algorithme ci-dessous :

Variables	A et B sont des nombres réels
Entrées	Saisir A et B
Traitement et sortie	**Si** $A < B$
	Alors afficher A
	Sinon afficher B
	Fin Si

1/ Pour chacune des entrées suivantes, **déterminer** la valeur affichée par l'algorithme de sortie :

a) A = 4 et B = 7 : ....

b) A = 12 et B = 9,2 : ....

2/ Que fait cet **algorithme** ?

....

....

<u>Exercice 1</u> :

La directrice d'un commerce de reprographie a créé un algorithme permettant de déterminer le montant payé par un client à partir du nombre de photocopies effectuées.

Variables	N est un entier, P est un nombre réel
Entrée	Saisir N
Traitement	**Si** $A < 30$
	Alors P prend la valeur $A \times 0{,}2$
	Sinon P prend la valeur $6 + (A - 30) \times 0{,}1$
	Fin Si
Sortie	Afficher P

1/ Quel est le prix **payé** par un client effectuant :

a) 28 photocopies ? ....

b) 30 photocopies ? ....

c) 52 photocopies ? ....

2/ **Déterminer** le prix unitaire des 30 premières photocopies et celui des photocopies suivantes.

....

....

3/ La commerçante décide de changer ses tarifs : les 20 premières photocopies seront facturées 0,25 euros et les suivantes 0,10 euros. **Modifier** l'algorithme.

<u>Exercice 2</u> :

Voici un algorithme :

Variables	A, B, C et D sont des réels
Entrées	Saisir A et B
Traitement	C prend la valeur $A - B$
	Si $C \leqslant 0$
	Alors Affecter à D la valeur $B - A$
	Sinon Affecter à D la valeur $A - B$
	Fin Si
Sortie	Afficher D

1/ Pour chacune des entrées suivantes, **déterminer** la valeur affichée en sortie :

a) A = 5 et B = 9 : ....

b) A = 2 et B = -2 : ....

c) A = -3 et B = -7 : ....

d) A = 8 et B = 2 : ....

2/ Que fait cet **algorithme** ?

....

....

....

<u>Exercice 3</u> :

Ecrire un algorithme permettant de calculer le prix à payer pour un utilisateur de téléphone portable lorsque celui-ci bénéficie d'une forfait de 2 heures pour 8 euros et où chaque minute au-delà du forfait est facturée 0,20 euro.

....

....

....

....

....

....

....

....

....

....

IV/ Instruction itérative : boucle « POUR »

Une **boucle « POUR »** est utilisée lorsque l'on veut **recommencer plusieurs fois** un même **bloc d'instructions**. Cette boucle est finie.

Voici sa structure :

On peut donner n'importe quel nom à la variable. A chaque tour de la boucle, **sa valeur augmente constamment de 1**, on appelle cela un « pas »

Pour *variable* variant de *début* à *fin* **faire**
| *instruction(s)*
Fin Pour

Exemple :

Traitement et sortie	**Pour** i variant de 1 à 5 **faire** | Afficher « bonjour ! » **Fin Pour**

Que permet d'obtenir **l'algorithme** suivant ?
....
....

Exercice 1 :

Voici un algorithme :

Variables	n et S sont des entiers
Entrée	Saisir n
Initialisation	S prend la valeur 0
Traitement	**Pour** i variant de 1 à n **faire** | S prend la valeur $S + i$ **Fin Pour**
Sortie	Afficher S

1. Exécuter cet algorithme avec $n = 5$ en entrée en remplissant un tableau du type suivant, où chaque colonne correspond à une étape :

i	1	2	3	4	5
S					

2. Que calcule cet algorithme ?
....
....
....

Exercice 2 :

Compléter la deuxième ligne de l'algorithme suivant pour qu'il affiche successivement :

a. 0, 7, 14, 21 et 28 ; **b.** 21, 28, 35, 42, 49, 56 et 63.

Variable	P est un entier
Traitement et sortie	**Pour** i variant de ... à ... **faire** | P prend la valeur $7 \times i$ | Afficher P **Fin Pour**

a) .... et ....
b) .... et ....

Exercice 3 :

Exécuter l'algorithme suivant avec $n = 6$ en entrée.

Variables	E et n sont des entiers
Entrée	Saisir n
Initialisation	E prend la valeur 1
Traitement	**Pour** i variant de 1 à n **faire** | E prend la valeur $E + 10^i$ **Fin Pour**
Sortie	Afficher E

On trouve ....

....
....
....

<u>Exercice 4</u> :
Ecrire un algorithme qui permette d'obtenir la somme des *n* premiers carrés $1^2 + 2^2 + 3^2 + ... + n^2$

....
....
....
....
....

<u>V/ Instruction itérative : boucle « TANT QUE »</u>

La **boucle « TANT QUE »** est une structure itérative avec **fin de boucle conditionnelle. Elle est utilisée quand on ne sait pas à l'avance combien d'itérations il y aura**.

<u>Voici sa structure</u> :

Tant que {*condition C vraie*} **faire**
| *instruction(s)*
Fin Tant que

<u>Remarque</u> : La condition C est testée en début de boucle, donc, si elle n'est pas vérifiée au départ, la boucle n'est jamais effectuée.

Exemple :

Voici un algorithme :

Variables	A, K et M sont des entiers
Entrée	Saisir A
Initialisation	K prend la valeur 0
	M prend la valeur 0
Traitement	**Tant que** .... **faire**
	| Afficher M
	| Affecter à K la valeur $K + 1$
	| M prend la valeur ....
	Fin Tant que

Compléter l'algorithme pour qu'il affiche tous les multiples entiers naturels de l'entier A strictement inférieurs à 1000.

....
....

<u>Exercice 1</u> :
On considère l'algorithme suivant :

Variable	U est un entier
Entrée	Saisir U
Traitement	**Tant que** $U > 7$ **faire**
	| U prend la valeur $U - 7$
	Fin Tant que
Sortie	Afficher U

1. Faire fonctionner cet algorithme avec $U = 25$.
2. Proposer deux nombres entiers différents qui donnent le nombre 5 en sortie.
3. Peut-on obtenir le nombre 11 en sortie ? Justifier.

....

<u>Exercice 2</u> :
Compléter l'algorithme suivant afin qu'il donne en sortie la plus petite valeur de l'entier N pour laquelle la somme des N premiers entiers naturels dépasse 10 000.

Variables	N et S sont des entiers
Initialisation	S prend la valeur 0
	N prend la valeur 0
Traitement	**Tant que** .... **faire**
	| N prend la valeur $N + 1$
	| S prend la valeur ....
	Fin Tant que
Sortie	Afficher N