

Bases d'algorithmique

Christophe ROSSIGNOL*

Année scolaire 2015/2016

Table des matières

1	Un peu de vocabulaire	2
1.1	Qu'est-ce qu'un algorithme?	2
1.2	Variable, affectation	2
2	Des structures importantes	3
2.1	l'instruction conditionnelle	3
2.2	La boucle itérative	5

Liste des algorithmes

1	Image par une fonction	3
2	Triangle rectangle en \mathcal{C}	4
3	Image par une fonction en tenant compte de l'ensemble de définition	4
4	Jeu de Pile ou Face	5
5	Table de multiplication	5
6	Calcul d'une somme d'entiers	6

*Ce cours est placé sous licence Creative Commons BY-SA <http://creativecommons.org/licenses/by-sa/2.0/fr/>

en préliminaire :

Activités : Activité 1¹ et 2² page 10 [TransMath]

1 Un peu de vocabulaire

1.1 Qu'est-ce qu'un algorithme ?

Définition : Un **algorithme** est une suite finie d'opérations élémentaires, à appliquer dans un ordre déterminé, à des données. Sa réalisation permet de résoudre un problème donné.

Exemples : suivre une recette de cuisine, suivre un plan, faire une division euclidienne à la main sont des exemples d'algorithme.

Remarques :

1. Un algorithme doit être lisible de tous. Son intérêt, c'est d'être codé dans un langage informatique afin qu'une machine (ordinateur, calculatrice, etc.) puisse l'exécuter rapidement et efficacement.
2. Les trois phases d'un algorithme sont, dans l'ordre :
 - (a) l'entrée des données
 - (b) le traitement des données
 - (c) la sortie des résultats

Exercices : 1, 3, 4, 5, 6, 8 page 11³ [TransMath]

1.2 Variable, affectation

Activités : Activité 1⁴ et 2⁵ page 12 [TransMath]

Définition : Lors de l'exécution d'un algorithme, on va avoir besoin de stocker des données, voire des résultats. Pour cela, on utilise des **variables**.
On attribue un nom à chaque variable.

Remarques :

1. Une variable est comme une boîte, repérée par un nom, qui va contenir une information. Pour utiliser le contenu de cette boîte, il suffit de l'appeler par son nom.
2. Dans l'écriture d'un algorithme, on prendra l'habitude de préciser dès le départ le nom des variables utilisées en indiquant leur type (nombre, chaîne de caractère, liste, etc.). Cette étape est appelée déclaration des variables.

Définition : Les instructions de base sur des variables sont les suivantes :

- la **saisie** : on demande à l'utilisateur de l'algorithme de donner une valeur à la variable ;
- l'**affectation** : le concepteur de l'algorithme donne une valeur à la variable. Cette valeur peut-être le résultat d'un calcul ;
- l'**affichage** : on affiche la valeur de la variable.

Exemple : L'algorithme 1 est un exemple d'algorithme calculant l'image d'un réel x par la fonction f :
 $x \rightarrow 3x^2 - 2x + 1$.

Remarques :

1. Dans l'algorithme 1, l'utilisateur *saisit* la variable x , alors que la variable y est *affectée* au cours du traitement.
2. On suivra toujours la structure de l'algorithme 1 lors de l'écriture d'algorithmes.

Exercices : 9, 10, 11 page 12⁶ – 13, 14 page 13⁷ [TransMath]

1. Labyrinthe.
2. Drôle de monstre.
3. Premiers algorithmes.
4. Programme de calcul.
5. Tracés dans un repère.
6. Premiers algorithmes.
7. Premières utilisations d'Algobox.

Algorithme 1 Image par une fonction

Variables :

x, y : nombres réels

Entrée :

Saisir x

Traitement :

y reçoit $3x^2 - 2x + 1$

Sortie :

Afficher y

2 Des structures importantes

2.1 l'instruction conditionnelle

Activités : Activité 1⁸ et 2⁹ page 14 [TransMath]

Définition : La résolution des certains problèmes nécessite la mise en place d'un **test** pour savoir si l'on doit effectuer une tâche.

Si la condition est remplie **alors** on effectue la tâche, **sinon** on effectue (éventuellement) une autre tâche.

Dans un algorithme, on code la structure du « Si... Alors.. Sinon » sous la forme suivante :

Si *condition*

Alors

Tâche 1

Tâche 2

...

Sinon

Tâche 1bis

Tâche 2bis

...

Fin Si

Remarques :

1. Il est important de respecter les espaces laissés au début de chaque ligne, car ils permettent une meilleure lisibilité de l'algorithme.
2. Le « Sinon » n'est pas obligatoire. S'il n'est pas présent, aucune tâche ne sera effectuée si la condition n'est pas remplie.

Exemples :

1. L'algorithme 2 permet de déterminer si un triangle ABC est rectangle en C .
2. L'algorithme 3 est un exemple d'algorithme calculant l'image d'un réel x par la fonction $f : x \rightarrow \frac{x+1}{x-1}$ en respectant son ensemble de définition.
3. L'algorithme 4 simule un jeu de pile ou face avec une pièce non truquée. « Pile » est représenté par le nombre 0 et « Face » par le nombre 1.

Exercices : 18, 19, 20, 21 page 15¹⁰ – 16, 17 page 15¹¹ [TransMath]

8. Mettre en évidence des conditions.
9. Théorème de PYTHAGORE
10. Instructions conditionnelles.
11. Cas où il y a plus de deux choix possibles.

Algorithme 2 Triangle rectangle en C

Variables :

AB, AC, BC, x , y : nombres réels

Entrée :

Afficher « Entrer la valeur de AB »

Saisir AB

Afficher « Entrer la valeur de AC »

Saisir AC

Afficher « Entrer la valeur de BC »

Saisir BC

Traitement :

x reçoit AB^2

y reçoit AC^2+BC^2

Si $x = y$

Alors

Afficher « Le triangle ABC est rectangle en C »

Sinon

Afficher « Le triangle ABC n'est pas rectangle en C »

Fin Si

Algorithme 3 Image par une fonction en tenant compte de l'ensemble de définition

Variables :

x , y : nombres réels

Entrée :

Saisir x

Traitement :

Si $x \neq 1$

Alors

y reçoit $(x+1)/(x-1)$

Afficher y

Sinon

Afficher « La valeur choisie n'est pas dans l'ensemble de définition »

Fin Si

Algorithme 4 Jeu de Pile ou Face

Variables :

choix, tirage : nombres réels

Entrée :

Saisir *choix*

tirage reçoit un nombre au hasard choisit dans l'ensemble $\{0; 1\}$

Traitement :

Si *choix=tirage*

Alors

Afficher « Gagné ! »

Sinon

Afficher « Perdu ! »

Fin Si

2.2 La boucle itérative

Activités : Activité 1¹² et 2¹³ page 16 [TransMath]

Définition : Lorsque l'on doit répéter un nombre de fois *connu à l'avance* la même tâche, on utilise une **boucle itérative** de la forme « **Pour.. allant de... à** ».

Dans un algorithme, cette structure est codée de la façon suivante :

Pour *variable* **allant de** *valeur_depart* **à** *valeur_fin* **faire**

tâche 1

tâche 2

...

Fin pour

La variable utilisée dans la boucle est appelée **compteur**. À chaque passage dans la boucle, sa valeur est automatiquement augmentée de 1.

Exemples :

1. L'algorithme 5 affiche la table de multiplication (de 0 à 10) d'un nombre entier donné.

Algorithme 5 Table de multiplication

Variables :

n, m, i : nombres entiers

Entrée :

Saisir *n*

Traitement :

Pour *i* allant de 0 à 10 faire

m reçoit $n \times i$

Afficher *n* ,« x », *i* , « = », *m*

Fin Pour

2. L'algorithme 6 affiche la somme de tous les entiers jusqu'à un entier donné.

Exercices : 23, 24, 26 page 17¹⁴ – 27, 28 page 17¹⁵ [TransMath]

-
12. Transmettre une information.
 13. Composer une table.
 14. Boucles itératives.
 15. Utilisation du langage Tortue.

Algorithme 6 Calcul d'une somme d'entiers

Variables :

S , i , n : nombres entiers

Entrée :

Saisir n

S reçoit 0

Traitement :

Pour i allant de 1 à n faire

S reçoit $S + i$

Fin Pour

Sortie :

Afficher S

Références

[TransMath] Transmath Seconde, Nathan (édition 2010).

2, 3, 5