

INTRODUCTION A LA PROGRAMMATION

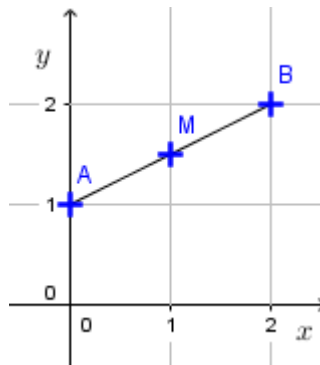
1°) Algorithme.

On appelle algorithme une succession d'instructions rédigées par un analyste pour un exécutant (souvent une machine).

Exemple :

Début.
 Tracer un repère orthonormé (O, I, J).
 Placer un point A (0 ; 1).
 Placer un point B (2 ; 2).
 Tracer [AB] .
 Placer M (1 ; 1,5).
 Fin.

Que fait cet algorithme ? Pour le comprendre, nous devons l'exécuter.
 L'algorithme réalise la figure suivante :

2°) Lecture, écriture.

En programmation, « lire » une valeur, ou « écrire » une valeur, est toujours utilisé dans le sens de la machine (ordinateur ou calculatrice).

Exemple :

Début.
 Ecrire (« Comment tu t'appelles ? »). *L'écran affiche « comment tu t'appelles ? »*
 Lire (*prenom*). *L'utilisateur écrit son prénom, mémorisé dans la variable prenom.*
 Ecrire (« Bonjour », *prenom*). *L'écran affiche bonjour suivi du prénom de l'utilisateur.*
 Fin.

Pour permettre de comprendre mieux un algorithme, ou un programme, on peut décrire en quoi consiste chaque instruction à côté, en commentaire.

3°) Variables.

Bien souvent, pour faire un algorithme qui puisse interagir avec l'utilisateur, nous aurons besoin de lire des informations données par l'utilisateur, et de stocker ces informations (comme le prénom dans l'exemple précédent).

Parfois, on devra faire des calculs internes avec ces valeurs. Ces calculs seront totalement invisibles par l'utilisateur : seul l'analyste programmeur a connaissance de comment fonctionne le programme.

Les valeurs que l'on doit stocker sont rangées dans des « variables » (un espace mémoire, une « boîte »).

Exemple : dans l'algorithme du prénom, on « range » le prénom de la personne dans une variable nommée *prenom*, ainsi, on peut utiliser cette valeur un peu plus tard, en disant bonjour à l'utilisateur.

Le nom de la variable peut n'avoir qu'une seule lettre.

Le nom de la variable doit permettre de retrouver facilement la valeur qui est dans la variable.

Selon le langage de programmation, les variables doivent être déclarées.

Exemple :

Variables :

a, b, c, d, e sont des nombres. } Variables

Début :

Lire a
Lire b } Initialisation
 $c \leftarrow a + b$
 $d \leftarrow a - b$ } Traitement
 $e \leftarrow c \times d + b^2$
Afficher e } Sortie

Fin.

Pour comprendre ce qu'exécute ce programme, on peut utiliser un tableau de variables :

Exécutons ce programme pour $a = 5$ et $b = 3$.

	a	b	c	d	e
Initialisation	5	3			
Traitement			8		
				2	
					25
Sortie					25

Tester ce programme avec les valeurs 7 et 1, puis avec les valeurs (-2) et 4.

Que fait, en réalité, ce programme ?

4°) Langages.

Les exemples d'algorithmes donnés précédemment sont en langage naturel.

Un algorithme doit ensuite être traduit en langage machine, afin d'être lu soit par un ordinateur, soit par la calculatrice.

Nous verrons plusieurs langages au cours de l'année :

- Scratch (utilisé pour l'introduction à certains concepts)
- Algobox (langage de programmation en français)
- Casio (pour ceux qui ont une Casio)
- TI (pour ceux qui ont une TI)

Il sera indispensable de connaître rapidement les exigences de langage de la calculatrice personnelle.

La règle suivante est toujours valable : UNE SEULE instruction par ligne.

Synthèse du langage de la calculatrice : voir en fin de livre.

Il faudra savoir :

- Créer un programme
- Editer un programme
- Exécuter un programme
- Supprimer un programme

- Afficher un texte
- Demander à l'utilisateur de donner la valeur d'une variable
- Affecter une valeur à une variable
- Afficher la valeur d'une variable

Exercices d'entraînement : (faites l'exercice puis traduisez l'algorithme de l'exercice en langage de votre calculatrice, programmez votre calculatrice, et testez le programme sur votre calculatrice)

A : 30 page 29 ; 32 page 30 ; 29 page 58 ;