

Algorithmes : Présentation

1. Notion d'algorithme

Définition

Un **algorithme** est une suite d'instructions détaillées qui, si elles sont correctement exécutées, conduit à un résultat donné.

Exemples

- une recette de cuisine, une notice de montage peuvent être considérées comme des algorithmes.
- la suite d'instructions suivantes :
 1. choisir un nombre entier
 2. le multiplier par lui-même
 3. énoncer le résultat obtenu

est un algorithme permettant d'obtenir le carré d'un nombre entier.

Remarque

Dans la définition précédente, « détaillées » signifie que les instructions sont suffisamment précises pour pouvoir être mises en oeuvre correctement par l'exécutant (homme ou machine)

2. Pseudo-code

Les instructions doivent être formulées dans un langage compréhensible par l'exécutant. Dans le cas d'un humain, il s'agira du langage courant (langue maternelle), ; dans le cas d'une machine, il faudra recourir à un langage de **programmation** (assembleur, basic, C, java, php ...).

En algorithmique, nous utiliserons un langage situé à mi-chemin entre le langage courant et un langage de programmation appelé *pseudo-code*. Il n'y a pas de norme concernant ce pseudo-code qui peut varier légèrement d'un enseignant à l'autre. Le but est surtout de mettre l'accent sur la logique de l'algorithme. L'avantage du pseudo-code est qu'il permet de rester proche d'un langage informatique sans qu'il soit nécessaire de connaître toutes les règles et spécificités d'un langage particulier.

3. Les variables

Un algorithme (ou un programme informatique), agit sur des nombres, des textes, ... Ces différents éléments sont stockés dans des **variables**. 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.



En informatique, les variables sont des emplacements réservés dans la mémoire de l'ordinateur auxquels on attribue une étiquette.

Définition

Déclarer une variable c'est indiquer le nom et le type (nombre texte, tableau,...) d'une variable que l'on utilisera dans l'algorithme.

La déclaration des variables se fait au début de l'algorithme avant la première instruction.

Remarques

- Pour reprendre l'image précédente, déclarer une variable consiste à « créer la boîte »
- Les principaux types de variables que nous utiliserons seront : entier, nombre (=réel), texte (=chaîne de caractères), tableau de nombres ou de textes, logique (=booléen -cf chapitre suivant)
- Lorsqu'on déclare une variable dans un programme informatique, l'ordinateur affecte une étiquette à une zone de mémoire et éventuellement réserve de l'espace pour le contenu de cette variable en fonction de son type.

Exemple

Dans notre pseudo-code, nous déclarerons les variables de la façon suivante :

```
variables  
x : nombre  
y : texte  
a, b, c : entiers
```

(Dans l'exemple précédent on définit 5 variables : x du type nombre (réel), y du type texte, et a, b et c de type entier.)

Nous distinguerons la déclaration des variables en plaçant le reste de l'algorithme entre les instructions « début algorithme » et « fin algorithme ».

Définition

Affecter une variable, c'est attribuer une valeur à cette variable. Si la variable contenait déjà une valeur, cette ancienne valeur est effacée.

Remarques

- Affecter une variable revient à « remplir la boîte »



- Dans notre pseudo-code, nous utiliserons l'expression «*prend la valeur*» pour l'affectation. Voici la déclaration et l'affectation de la variable x :

```
variables
x : entier
début algorithme
x prend la valeur 5
fin algorithme
```

- on ne peut affecter à une variable qu'une valeur du type qui a été défini lors de la déclaration. Le code suivant est incorrect (le // indique le début d'un commentaire):

```
variables
x : entier
début algorithme
x prend la valeur "bonjour" // Erreur! x est de type entier !
fin algorithme
```

- Les textes (ou chaînes de caractères) doivent être entourés d'apostrophes afin de ne pas être confondus avec des noms de variables.

Il est possible d'affecter à une variable le contenu d'une autre variable ou le résultat d'un calcul. Le contenu de l'autre variable n'est alors pas modifié. Par exemple :

```
variables
x, y, z : entiers
début algorithme
x prend la valeur 5
```

```
y prend la valeur x
z prend la valeur x+y+1
fin algorithme
```

A la fin de cet algorithme, x et y contiennent la valeur 5 et z la valeur 11 (= 5+5+1).

4. Les instructions d'entrée-sortie

Faire effectuer un calcul à une machine c'est bien... Mais il faut au moins être capable d'entrer des valeurs et il faut aussi que la machine puisse afficher un résultat !

Les instructions qui permettent de « dialoguer » avec une machine s'appellent les instructions « **d'entrée/sortie** » ou de « **lecture/écriture** »

Lecture

Dans notre pseudo-code nous utiliserons l'instruction ***lire*** (ou *entrer*, ou *saisir*, etc.) **suivie du nom d'une variable** pour pouvoir saisir une valeur (en anglais cette instruction se nomme généralement *input*).

Lorsqu'elle rencontre une telle instruction, la machine s'**arrête** et **attend que l'utilisateur entre une valeur**. Une fois la valeur saisie, la machine **affecte la valeur saisie** à la variable dont le nom figure après *lire*. Ensuite, elle passe à l'**instruction suivante**.

```
variables
x : entier
début algorithme
lire x
y prend la valeur 2*x
fin algorithme
```

Cet algorithme demande d'**entrer** un nombre entier, **stocke** la valeur de ce nombre dans la variable x, puis **calcule le double** du nombre entré et **affecte ce double à la variable y**.

Le résultat n'est pas affiché (d'où le paragraphe suivant...)

Remarque : Dans un véritable programme, il faudrait vérifier que la valeur entrée est bien du type désiré (ici un entier). Dans un algorithme, on n'écrit pas cette vérification.

Ecriture

Dans notre pseudo-code nous utiliserons l'instruction ***afficher*** suivie du nom d'une variable ou d'une constante (nombre, texte ...) pour afficher une valeur (on peut également utiliser « ***écrire*** » ou « ***print*** » en anglais).

Pour afficher un texte on utilise des **guillemets** (simples ou doubles) :

```
...
afficher 'Ce texte sera affiché'
...
```

Il est fréquent d'afficher un texte pour donner des **consignes** ou des **informations** à l'utilisateur.

Pour afficher le contenu d'une variable on fait suivre « *afficher* » du nom de la variable **sans guillemet** :

```
...
afficher x
...
```

L'algorithme suivant calcule puis affiche l'âge qu'aura une personne en 2100 :

```
variables
annee, age, age_en_2100 : entiers \
début algorithme
afficher "Entrer l'année actuelle"
lire année
afficher "Entrer votre âge"
lire âge
âge_en_2100 prend la valeur âge + 2100 - année
afficher "En 2100, vous aurez ", âge_en_2100, " ans."
fin algorithme
```

Si l'utilisateur entre « 2014 » comme année et « 16 » comme âge, l'algorithme affichera :

En 2100, vous aurez 102 ans.

