

Définition 1 (Wikipédia) **L'algorithmique** est l'ensemble des règles et techniques qui sont impliquées dans la définition et la conception d'algorithmes. Un **algorithme** est une suite finie et non-ambiguë d'opérations ou d'instructions permettant de résoudre un problème.

Définition 2 Un **programme** (informatique) est la traduction d'un algorithme dans un langage de programmation donné.

Distinction. Un algorithme peut être écrit dans un langage proche du langage naturel : c'est ce qu'on appelle le **pseudo-langage**. Un programme doit être écrit dans un langage compris par la machine sur laquelle il est exécuté. C'est le langage de programmation. **En TD** on utilise **uniquement** le pseudo-langage.

Définition 3 Dans ce module nous allons aborder la **programmation impérative**. Il s'agit d'un paradigme de programmation (grossièrement une façon de programmer) dans laquelle l'ordinateur exécute les instructions **séquentiellement**, l'une après l'autre. Cela veut dire que **l'ordre** dans lequel sont écrites les instructions est (très) **important**.

Déclaration. Tout objet manipulé dans un algorithme doit être **déclaré avant de pouvoir être utilisé**. Un objet est caractérisé par un identifiant (c'est-à-dire un nom) **unique**. Il existe deux catégories d'objets manipulés dans les algorithmes : les constantes et les variables.

Définition 4 Une **constante** correspond à un objet dont la valeur ne change pas pendant le déroulement de l'algorithme. La constante peut être implicite (par exemple 1 ou 2.5) ou explicite, c'est-à-dire définie par le programmeur (par exemple $MAX = 100$). L'identifiant d'une constante est généralement composé uniquement de majuscules. Lorsque l'identifiant d'une constante contient plusieurs mots, on les sépare par le caractère '_'.

Définition 5 Une **variable** est un objet dont la valeur peut changer en cours de traitement. La valeur de la variable appartient à un ensemble ou à un domaine de définition : c'est la notion de **type**. L'identifiant d'une variable commence généralement par une minuscule. Lorsque l'identifiant contient plusieurs mots, on les sépare par le caractère '_'.

Définition 6 Le **type** d'une variable définit l'ensemble des valeurs que peut prendre la variable **et** l'ensemble des actions (ou opérations) que l'on peut effectuer sur la variable.

Les types simples :

- le type **Booléen** : {faux, vrai}
- le type **Caractère** : { 'a', 'b', ..., 'z', 'A', 'B', ..., 'Z', '0', '1', ..., '9', ',', '?', ... }
- le type **Entier** : c'est l'équivalent de l'ensemble $\mathbb{Z}$ en mathématique.
- le type **Réel** : c'est l'équivalent de l'ensemble $\mathbb{R}$ en mathématique.

Entités intervenant dans un algorithme :

- le mot **ordinateur** désigne la machine sur laquelle le programme informatique est exécuté.
- le mot **utilisateur** désigne la personne qui se trouve face à l'ordinateur.
- le **clavier** est un outil permettant une interaction entre l'ordinateur et l'utilisateur. Il permet de **lire** (ou saisir) des informations entrées par l'utilisateur.
- l'**écran** est un outil permettant également une interaction entre l'ordinateur et l'utilisateur. Il permet d'**écrire** (ou afficher) des informations ou des messages pour l'utilisateur.

Définition 7 Un algorithme est toujours composé d'un en-tête et d'un corps. L'**en-tête** regroupe le nom de l'algorithme et les données qu'il utilise. Le **corps** est la suite d'instructions à exécuter séquentiellement. Il correspond au traitement à réaliser. Il commence (en pseudo-langage) par le mot clé **début** et se termine par le mot clé **fin**.

Définition 8 Une **primitive** est une action de base du langage algorithmique. Elle correspond à une action qui peut être exécutée directement par l'ordinateur. Il existe 3 primitives : **l'affectation** ; **la lecture** ; **l'écriture**. L'**affectation** permet d'attribuer (ou affecter) une valeur, ou le résultat d'une expression, à une variable. Elle a pour but de **modifier** la valeur de la variable. La **lecture** permet à l'utilisateur de saisir une donnée (valeur, texte) au clavier. La donnée est récupérée dans l'algorithme et elle est affectée à une variable. Cela implique d'avoir déclaré une variable de même type que ce que l'on va lire. L'**écriture** permet à l'algorithme (l'ordinateur) d'afficher une donnée (valeur, texte) à l'écran.

Pseudo-langage. Un algorithme écrit en pseudo-langage respecte un ensemble de **règles**. Celles-ci reposent en partie sur la définition de quelques mots clés.

- la déclaration d'une constante se fait grâce au mot clé **CONST**, et se fait avant le début de l'algorithme
- la première ligne de l'algorithme est de la forme : **Programme nom_du_programme**
- la déclaration d'une variable se fait grâce au mot clé **VAR**
- le corps de l'algorithme commence avec le mot clé **début** et se termine avec le mot clé **fin**
- l'affectation est représentée par le symbole $\leftarrow$ (**Attention** : ne pas confondre avec $=$)
- l'instruction de lecture est traduite par le mot clé **Lire**
- l'instruction d'écriture est traduite par le mot clé **Écrire**
- La fin d'une instruction est marquée par un point-virgule ;



Ressources complémentaires

En complément de cette fiche vous trouverez sur Moodle deux activités à réaliser **avant** la première séance de travaux dirigés contenant 2 vidéos et de petits exercices pour renforcer votre compréhension. Un QCM est à valider après avoir réalisé les 2 activités (et toujours avant la première séance de travaux dirigés).



Exercices à faire en auto-formation

Les exercices ci-dessous sont des exercices à faire en autonomie, pour s'assurer d'avoir assimiler les notions de base présentées dans la fiche et dans les activités proposées sur Moodle.

Exercice 1 :

Quelles seront les valeurs affichées après exécution des instructions suivantes :

```
Programme ex01a
  VAR a, b : Entier;
début
  a  $\leftarrow$  8 ;
  b  $\leftarrow$  a * 4 ;
  a  $\leftarrow$  1 ;
  b  $\leftarrow$  b - 4 * a ;
  écrire(a, b) ;
fin
```

```
Programme ex01b
  VAR a, b, c : Entier;
début
  a  $\leftarrow$  2 ;
  b  $\leftarrow$  4 ;
  a  $\leftarrow$  1 + b ;
  c  $\leftarrow$  a + b ;
  écrire(a, b, c) ;
fin
```

```
Programme ex01c
  VAR a, b : Entier;
début
  a  $\leftarrow$  4 ;
  b  $\leftarrow$  33 ;
  a  $\leftarrow$  b ;
  b  $\leftarrow$  a ;
  écrire(a, b) ;
fin
```

Exercice 2 :

Quelles seront les valeurs des variables à la fin des algorithmes suivants ?

```
Programme ex02a
  VAR a, b, c : Entier;
début
  a  $\leftarrow$  30 ;
  b  $\leftarrow$  a DIV 5 ;
  c  $\leftarrow$  a - b * 5 ;
  écrire(a, b, c) ;
fin
```

```
Programme ex02b
  VAR a, b, c : Entier;
début
  a  $\leftarrow$  15 ;
  b  $\leftarrow$  a DIV 2 ;
  c  $\leftarrow$  a - b * 2 ;
  écrire(a, b, c) ;
fin
```

Exercice 3 :

Simuler le déroulement des actions suivantes pour déterminer les valeurs affichées à l'écran, en fonction de celles saisies initialement.

```
Programme exo3a
  VAR A, B : Entier;
début
  Ecrire("Entrer un entier entre 1 et 9 :");
  Lire(A);
  A ← A * 37;
  B ← A;
  A ← A + A;
  A ← A + B;
  écrire(A);
fin
```

```
CONST PI = 3.1415;
Programme exo3b
  VAR a, b : Réel;
début
  Lire(a);
  b ← a;
  a ← 2 * PI * a;
  écrire(a, b);
fin
```

Exercice 4 :

Compléter l'algorithme suivant pour qu'il soit correct :

```
Programme exo5
  VAR ... : ...
début
  ...
  n ← n + m;
  p ← n DIV 2;
  m ← n + p;
  écrire(n, m, p);
fin
```

Exercice 5 :

Un objet en forme de bille est lâché depuis la fenêtre d'un immeuble. Au bout de t secondes, l'objet est descendu de $h = (1/2)gt^2$ mètres, avec $g = 9.81m.s^{-2}$.

1. Écrire un algorithme qui calcule et affiche la hauteur descendue au bout d'un temps t saisi par l'utilisateur.
2. Écrire un algorithme qui calcule et affiche la durée totale de la chute, en fonction de la hauteur totale h lue au clavier.

Exercice 6 :

Écrire un algorithme qui calcule et affiche l'aire d'un triangle dont la base et la hauteur sont demandées à l'utilisateur.