

I Algorithme

I.1 Définition

Définition : Un algorithme est une suite d'instructions élémentaires, qui une fois exécutée correctement, conduit à un résultat donné.

I.2 Quelques remarques

Un algorithme possède un début et une fin et doit se terminer en un temps fini.

Les instructions composant un algorithme sont suffisamment simples pour être compréhensibles par l'exécutant (homme ou machine). On les appelle alors des **instructions élémentaires**.

« **exécutée correctement** » signifie que l'exécutant (homme ou machine) doit appliquer strictement les instructions les unes après les autres, sans se poser la question de la justesse du résultat, ni de la façon dont il est obtenu. Si l'algorithme est bien fait, le résultat est correct dans tous les cas de figure.

I.3 Exemples de la vie courante

Exemple 1 Une recette de cuisine ;

Exemple 2 Une notice de montage ;

Exemple 3 Un itinéraire ;

Exemple 4 À l'école élémentaire, la multiplication de deux nombres entiers, au collège le protocole de construction de la médiatrice d'un segment. etc ...

I.4 Règles à suivre pour concevoir un algorithme

liste non exhaustive

Règle 1 Vérifiez sur des cas simples que vous avez compris le problème à résoudre et que vous savez le faire à la main.

Règle 2 Décomposez votre résolution du problème en actions de plus en plus simples ... jusqu'à atteindre un niveau élémentaire que l'exécutant comprendra.

Règle 3 Choisissez intelligemment les noms de variable¹ que vous serez amenés à utiliser dans l'algorithme.

Règle 4 Mettre des commentaires² dans l'algorithme.

Règle 5 Testez votre algorithme à la main sur des cas simples³.



1. dès que l'on a besoin de stocker une information (numérique, texte, etc ...) dans un algorithme, on utilise une **variable**. Une variable est une "boîte", que le programme (l'ordinateur) va repérer par une "étiquette". Pour avoir accès au contenu de la boîte, il suffit de la désigner par son étiquette. La première chose à faire avant de pouvoir utiliser une variable est de créer la boîte et de lui coller une étiquette. Ceci se fait tout au début de l'algorithme, avant même les instructions proprement dites. C'est ce qu'on appelle la **déclaration des variables**. Les noms de variables ne doivent pas comporter d'espace et ne peuvent pas commencer par un chiffre. D'une façon générale, on se limitera aux caractères alphanumériques non accentués et au tiret de soulignement. La bonne lisibilité d'un algorithme étant fondamentale, il est important de choisir des noms de variable reflétant le rôle joué par la variable en question.
2. cela n'a aucune utilité pour le bon fonctionnement de l'algorithme, mais le rend plus lisible et plus compréhensible.
3. le concepteur doit être capable de simuler le comportement d'un exécutant.

II Algorithmique et Langage

Une façon de s'adresser à un ordinateur pour lui faire remplir une tâche est de lui parler dans son langage primaire : le langage machine.

C'est pour cette raison que se sont développés des langages appelés **langages de programmation** qui sont en quelque sorte des **interprètes** traduisant des instructions claires en langage machine.

Lorsqu'il est traduit dans un langage de programmation, l'algorithme s'appelle un programme et son concepteur le programmeur ou développeur s'il s'agit de projets plus sophistiqués.

Le but de ce cours est d'apprendre à écrire des algorithmes, c'est à dire à décomposer une action compliquée en instructions simples. Cet exercice est a priori totalement indépendant d'un langage de programmation et reste un travail de papier. Nous écrirons nos algorithmes dans un langage libre¹ et, pour les tester sur une machine, nous les traduirons dans un langage de programmation : il s'agira cette année de PYTHON².



1. Ceci sera précisé ultérieurement.

2. Télécharger la version 3.4.3 : <https://www.python.org/downloads/>

III Instructions élémentaires

III.1 Structure d'un algorithme

Nom de l'algorithme (Choisir un nom évoquant le résultat escompté)

Déclaration des variables (on liste les variables à utiliser)



Début de l'algorithme (corps de l'algorithme)

Suite d'instructions élémentaires

...

Fin de l'algorithme



III.2 Utilisation des variables

III.2.1 Affectation d'une variable

Le fait d'attribuer un contenu à une variable s'appelle l'affectation. Dans l'écriture d'un algorithme en **langage libre**, on la symbolisera par \leftarrow selon la syntaxe :

NomVariable \leftarrow valeur

signifiant que la quantité située à droite de la flèche est évaluée, puis stockée dans la variable nommée NomVariable, remplaçant ainsi son ancien contenu.

Exemple 1

```
N ← 7
message ← "bonjour"
test ← vrai
A ← B
resultat ← nb1+nb2
```

Met la valeur 7 dans la variable N Met dans la variable message la chaîne de caractères bonjour Met la valeur vrai dans la variable booléenne test Met dans la variable A le contenu de la variable B Additionne les contenus des variables nb1 et nb2 et met le résultat dans resultat
--

III.2.2 Types de variable

On distingue pour commencer 4 types de variables : entier, réel, chaîne de caractères, booléen.

Les deux premiers types servent à manipuler les nombres entiers ou « à virgule » (\mathbb{Z} ou \mathbb{R}). Les chaînes de caractères sont conventionnellement écrites entre guillemets dans un algorithme.

Une variable de type booléen ne peut accueillir que l'une des deux valeurs **Vrai** ou **Faux**.

Les affectations opérées par l'algorithme doivent évidemment être cohérentes avec les types déclarés. Par exemple, si les variables **A** et **B** sont déclarées entières, les instructions $A \leftarrow 3.7$ ou $A \leftarrow B/2$ sont des erreurs. De même, si **A** une variable de type entière et **B** de type réelle, l'affectation $A \leftarrow B$ engendre une erreur, alors que $B \leftarrow A$ est une instruction correcte.

Il convient de faire la différence entre "3.14" (chaîne de caractères) et 3.14 (nombre réel) ou entre 3.0 (nombre réel) et 3 (nombre entier).

Avant leur utilisation, les **variables** doivent toujours être **initialisées** notamment par l'affectation d'une valeur.

III.3 LECTURE et ÉCRITURE

Un algorithme doit communiquer avec son utilisateur.

On écrira, en langage libre, la lecture d'une donnée saisie au clavier par l'utilisateur et son stockage dans une variable :

$$\text{NomVariable} \leftarrow \text{saisir}() \quad \text{ou simplement} \quad \text{saisir}(\text{NomVariable})$$

Important : En arrivant sur cette instruction, le programme se met en attente d'une entrée. L'utilisateur tape quelque chose au clavier et le valide, provoquant le stockage de la valeur tapée dans la variable **NomVariable** et la reprise de l'algorithme à l'instruction suivante.

L'écriture en langage libre d'un résultat ou d'un message se fait grâce à l'instruction **afficher(...)**.

Attention de ne pas confondre les instructions **afficher("Bonjour")** et **afficher(Bonjour)** : la première écrira la chaîne de caractères *Bonjour* à l'écran alors que la seconde écrira le contenu de la variable **Bonjour**. (*curieux nom de variable d'ailleurs !!!*)

On peut afficher plusieurs informations à la suite en les séparant par des virgules à l'intérieur de l'instruction **afficher**.

Exemple 2 La séquence d'instructions suivantes :

```

...
A ← saisir("Saisir une valeur de A :□")
B ← saisir("Saisir une valeur de B :□")
afficher("A□vaut□",A,"□,□B□vaut□",B,"□et□leur□somme□vaut□",A+B)   (□ désignant un caractère espace)
...

```

permettra l'affichage à l'écran de : ...

IV Des exercices

EXERCICE 1 : Un algorithme écrit en langage "libre"

<pre> Déclaration des variables A est un entier B est un entier Début A ← 1 B ← 2*(A+3) A ← 3 Fin </pre>	En Python,
--	------------

Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

EXERCICE 2 : Un algorithme écrit en langage "libre"

<pre> Déclaration des variables A, B et C sont des entiers Début 1 A ← 1 2 B ← 2 3 C ← B+3 4 A ← A+1 5 C ← C+A 6 afficher("les trois entiers sont : ",A," ",B," ",C) 7 afficher("leur somme est : ",A+B+C) Fin </pre>	En Python,
--	------------

Remplir le tableau des valeurs prises par les variables au fur et à mesure que l'algorithme se déroule.

<i>n</i> ^o de ligne							
<i>A</i>							
<i>B</i>							
<i>C</i>							

EXERCICE 3 : Un algorithme écrit en langage "libre"

<pre> Déclaration des variables prenom est une chaîne de caractères Début prenom ← saisir() Afficher("Bonjour□") Afficher(prenom) Fin </pre>	En Python,
--	------------

Que fait cet algorithme ?

EXERCICE 4 : Écrire un algorithme en langage "libre" qui demande un nombre à l'utilisateur et affiche "ce nombre multiplié par 7 vaut : "