



Université Félix Houphouët-Boigny
(UFHB)

Cours d'algorithmie avancée L3-Info

Enseignant :

Dr GBAME Gbede Sylvain

Assistant, enseignant chercheur à l'UFHB

ggamegbedesylvain@gmail.com



MODULE : Algorithme avancée

PLAN

Chapitre 0 : Généralité sur l'algorithme avancées

Chapitre 1 : variables, types, instructions élémentaires et expressions

Chapitre 2 : Les structures conditionnelles

Chapitre 3 : Les structures répétitives

Chapitre 4: Les tableaux

Chapitre 5: Les tris

Chapitre 6: Les sous-algorithmes

Chapitre 1

Variables, types, instructions et expressions

Chapitre 1 : Variables, types, instructions et expressions

1.1 La structure d'un algorithme

1. Un exemple d'algorithme

Problème : écrire un algorithme qui *calcule la factorielle d'un nombre* saisi au clavier. Différents algorithmes résolvent ce problème. En voici un :

Dans ce chapitre, on ne s'intéressera qu'à la structure de cet algorithme.

Chapitre 1 : Variables, types, instructions et expressions

1.1 La structure d'un algorithme

Algorithme CalculFactorielle

Var nombre, sauvegarde, resultat : entier

Début

 écrire ("Calcul de la factorielle d'un nombre")

 Répéter

 écrire ("Donnez un entier naturel : ")

 lire (nombre)

 Jusqu'à (nombre \geq 0)

 sauvegarde \leftarrow nombre

 Si ((nombre = 0) OU (nombre = 1))

 Alors resultat \leftarrow 1

 Sinon resultat \leftarrow 1

 TantQue (nombre > 1) Faire

 resultat \leftarrow nombre * resultat

 nombre \leftarrow nombre - 1

 FinTantQue

 FinSi

 écrire ("La factorielle de ", sauvegarde, "est ", resultat)

Fin

Chapitre 1 : Variables, types, instructions et expressions

1.1 La structure d'un algorithme

2. Les principales parties d'un d'algorithme

Un algorithme comporte trois parties : l'en-tête, la zone des déclarations et le corps.

2.1. L'en-tête

L'**en-tête** est annoncé par le mot réservé (ou mot-clé) **Algorithme** suivi du nom que l'on a choisi pour désigner l'algorithme. Le nom d'un algorithme est aussi appelé identificateur.

Le choix de l'identificateur est libre, mais il doit respecter certaines règles. Ces règles seront présentées dans la leçon suivante.

L'**en-tête** sert à nommer un algorithme de sorte à ne pas réécrire cet algorithme à nouveau lorsque l'on en aura besoin.

L'algorithme de l'exemple a pour en-tête : *Algorithme CalculFactorielle*

Chapitre 1 : Variables, types, instructions et expressions

1.1 La structure d'un algorithme

2.2. La zone des déclarations

La **zone des déclarations** est l'espace réservé pour lister les "ingrédients" dont on a besoin pour résoudre un problème. Ces ingrédients sont appelés objets algorithmiques.

Ce sont : les **constantes**, les **types**, les **procédures**, les **fonctions**, et les **variables**.

La zone des déclarations est délimitée par les mots réservés Algorithme et Début.

Dans l'algorithme de l'exemple, trois **variables** (nombre, sauvegarde et resultat) ont été déclarées.

Les **variables** forment une catégorie d'objets algorithmiques.

Le mot réservé **Var** (ou **Variables**) sert à déclarer les variables.

Chapitre 1 : Variables, types, instructions et expressions

1.1 La structure d'un algorithme

2.3. Le corps

Le corps d'un algorithme est la zone située entre les mots réservés **Début** et **Fin**.

On y écrit toutes les instructions qui permettent de résoudre le problème.

Le mot réservé **Début** indique le début de la suite d'instructions, le mot réservé **Fin** marque, à la fois, la fin de la suite d'instructions, et la fin de l'algorithme.

Donc il n'y a plus d'instruction après le mot **Fin**

Chapitre 1 : Variables, types, instructions et expressions

1.1 La structure d'un algorithme

3. Formalisme (ou schéma général) d'un algorithme

```
Algorithme < Nom_de_l'algorithme >   // En-tête de l'algorithme
// Zone des déclarations - Liste des objets algorithmiques
Constantes
...
Types
...
Procédures
...
Fonctions
...
Variables
...
// Corps de l'algorithme
Début
    < instruction1 ou action1 >
    < instruction2 ou action2 >
        .
        .
        .
    < instruction_n ou action_n >
Fin
```

Chapitre 1 : Variables, types, instructions et expressions

1.1 La structure d'un algorithme

Remarques

➤ **au sujet du fonctionnement d'un algorithme** : de manière générale, un algorithme reçoit des données, effectue des traitements sur ces données, et retourne des résultats.

➤ **au niveau syntaxique** : On peut insérer des commentaires dans un algorithme. Ils permettent d'annoncer les intentions de celui qui écrit l'algorithme. Bien que facultatifs, ces commentaires ne doivent pas se confondre aux instructions et aux mots réservés de l'algorithme.

Pour ce faire, on doit les délimiter avec les signes `/*` et `*/` lorsque le commentaire s'étend sur plusieurs lignes, ou avec le signe `//` lorsque le commentaire tient sur une seule ligne.

➤ **au sujet de la déclaration des objets algorithmiques** : L'utilisation d'objets algorithmiques dépend du problème à résoudre. Dans l'exemple, on n'a utilisé qu'une seule catégorie d'objets : les variables. Et dans cette catégorie, on a déclaré trois objets.

Chapitre 1 : Variables, types, instructions et expressions

1.2 Les Objets algorithmiques

1. Notion d'objet

Un objet doit être impérativement déclaré avant d'être utilisé dans un algorithme. On donne alors ses caractéristiques : son identificateur, son type et si possible sa valeur.

1.1. L'identificateur

L'identificateur est le nom choisi pour l'objet. Il se présente sous la forme d'une suite de caractères alphanumériques (alphabétiques et/ou numériques) sans espace. Il commence toujours par une lettre. Il est recommandé de choisir un nom qui a un lien sémantique avec le contenu de l'objet ou avec le rôle de l'objet.

Les caractères accentués et les signes de ponctuation ne sont pas autorisés.

Par exemples : nombre, resultat sont acceptés tandis que 1nombre, résultat sont refusés.

1.2. Le type

Avant d'appliquer un quelconque traitement à un objet, il faut d'abord connaître son type. Un objet est soit du type numérique, soit du type chaîne de caractères, soit du type booléen.

Chapitre 1 : Variables, types, instructions et expressions

1.2 Les Objets algorithmiques

👉 **Exemple** : On considère deux objets a et b contenant des données. S'il est possible d'additionner ces données, alors ces données sont des nombres. Il faut donc nécessairement que les deux objets qui les contiennent soient du type numérique. La notion de type d'un objet sera complètement exposée à la leçon suivante.

1.3. La valeur

Le traitement d'un objet porte sur la valeur de la donnée que cet objet contient.

Il y a deux cas :

- si cette valeur est modifiable, alors l'objet est une variable,
- si cette valeur n'est pas modifiable, alors l'objet est une constante.

Voici les formalismes respectifs de déclaration d'une variable et d'une constante :

Var < identificateur de la variable > : < **type** de la variable >

Const < identificateur de la constante > ← < valeur affectée à la constante >

Chapitre 1 : Variables, types, instructions et expressions

1.2 Les Objets algorithmiques

Exemple :

Dans l'algorithme précédent, deux variables ont été déclarées : *nombre*, *sauvegarde* et *resultat*. La variable appelée *nombre* reçoit le nombre dont on veut calculer la factorielle ; la variable nommée *sauvegarde* reçoit une copie du contenu de la variable *nombre*, et enfin la variable appelée *resultat* reçoit le résultat du calcul.

Elles sont toutes les trois de type numérique et précisément de ***type entier***.

D'où la déclaration : **Var** nombre, sauvegarde, resultat : entier.

Chapitre 1 : Variables, types, instructions et expressions

1.2 Les Objets algorithmiques

☞ Remarques

❖ à propos de la notion de variable : la notion de variable en informatique semble identique à celle utilisée en mathématiques. Il y a une grande différence. En effet :

➤ *la variable informatique*, qui est une case mémoire, à tout instant donné du déroulement de l'algorithme, ne peut contenir qu'une seule valeur, bien que celle-ci puisse changer du fait de certaines instructions de l'algorithme.

➤ *la variable mathématique* se comporte différemment selon les situations. Par exemples :

- dire que x appartient à \mathbb{R} , signifie que la variable x peut désigner n'importe quel nombre réel ;
- dire que n est un nombre premier, signifie que la variable n peut désigner n'importe quel nombre premier ;
- dans l'équation x appartenant à \mathbb{R} , $ax^2 + bx + c = 0$, x désigne "simultanément" deux racines réelles, si elles existent.

Chapitre 1 : Variables, types, instructions et expressions

1.2 Les Objets algorithmiques

❖ à propos des caractéristiques d'un objet :

- lors de la déclaration d'un objet, on précise son *identificateur*, son *type* et sa *valeur* si possible.
- lorsque l'on déclare un objet, on donne, en réalité, l'ordre au système d'exploitation de réserver un espace dans la mémoire vive de l'ordinateur, pour la mémorisation de cet objet. Dès que cet ordre est exécuté, une quatrième caractéristique s'ajoute : l'**adresse**.

Une **adresse** est la valeur numérique qui permet de localiser une donnée dans la mémoire vive.

👉 **Par exemple**, à la déclaration des variables *nombre*, *sauvegarde* et *resultat*, le système d'exploitation effectue une allocation de mémoire pour leur réserver des espaces dans la mémoire vive. Cependant, les adresses de ces espaces sont inconnues du concepteur.

- Chaque objet algorithmique est déclaré à l'aide d'un mot réservé propre.
- Les objets algorithmiques types, procédures et fonctions seront étudiés plus loin

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

1. Rappel

Une variable sert à stocker une donnée dans la mémoire vive de l'ordinateur. La variable peut mémoriser la donnée si elle est du même type que cette donnée.

2. Définition

Le type d'un objet est défini :

- d'une part, par l'ensemble des valeurs que cet objet peut prendre ;
- et d'autre part, par l'ensemble des opérateurs que l'on peut appliquer à ces valeurs.

3. Les principaux types de base

On distingue trois principaux types de base (ou types élémentaires ou types simples) : le type numérique, le type chaîne de caractères et le type booléen.

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

3.1. le type numérique

Le *type numérique* regroupe tous les *nombre*s : les *entiers* et les *réels*.
On déclare alors une donnée numérique soit dans le *type réel*, soit dans le *type entier*.

Le type numérique supporte les *opérateurs arithmétiques*, les *fonctions trigonométriques* et les *fonctions mathématiques*. Les opérations les plus courantes sont regroupées dans le tableau ci-dessous :

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

3.1. le type numérique

Opérations	Opérateurs et Fonctions
addition	+
soustraction	-
opposé de	-
multiplication	*
division (ou <i>division décimale</i>)	/
division entière (ou <i>division euclidienne</i>)	DIV
reste d'une division entière (ou <i>modulo</i>)	MOD
élévation à la puissance	^ ou ↑
partie entière d'un nombre	ENT (x)
valeur absolue d'un nombre	ABS (x)
logarithme népérien d'un nombre	ln (x)
logarithme à base 10 d'un nombre	log10 (x)
sinus, cosinus, tangente d'un nombre	sin (x), cos (x), tan (x)
arc sinus, arc cosinus, arc tangente d'un nombre	arcsin (x), arcos (x), arctan (x)
sinus, cosinus et tangente hyperboliques d'un nombre	sinh (x), cosh (x), tanh (x)

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

Opérateurs	Signification
=	est égal à
>	est strictement supérieur à
<	est strictement inférieur à
>= ou ≥	est supérieur ou égal à
<= ou ≤	est inférieur ou égal à
<> ou ≠	est différent de

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

3.2. le type chaîne de caractères (ou type chaîne)

Une **chaîne de caractères** est une suite finie de caractères.

Elle peut être :

- une chaîne vide ;
- ou un caractère suivi d'une chaîne de caractères.

Un **caractère** peut être :

- une lettre de l'alphabet (minuscule ou majuscule) ;
- un chiffre ;
- ou un symbole ou un signe quelconque : signe d'opérations (+ - × /), signe de ponctuation (! ? ; :) ou d'autres signes (§ £ ~ # *).

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

On retient alors :

Une chaîne de caractères est une suite finie de caractères regroupés dans une seule variable. Le type chaîne de caractères supporte les ***opérateurs de comparaison*** et les ***instructions de manipulation de chaînes de caractères***.

Une chaîne de caractères est délimitée par des griffes (" ") tandis qu'un caractère est délimité par des apostrophes (' ').

🔑 **Exemples** : "Calcul de la factorielle d'un nombre", 'D', 'd'.

Le type chaînes de caractères supporte les mêmes opérateurs de comparaison que ceux supportés par le type numérique.

Il existe des tables (telle que la table ASCII, page 10) qui regroupent tous les caractères utilisés en informatique. Dans ces tables, chaque caractère est lié à un code numérique, c'est-à-dire un nombre, qui le représente dans l'ordinateur. Ce code numérique permet d'établir une relation d'ordre entre les chaînes de caractères.

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

☞ Exemples :

- 'a' > 'A' car dans la table ASCII, le code numérique lié à 'a' est supérieur au code numérique lié à 'A'. En effet, $97 > 65$
- 'A' < 'Z' car, selon la table ASCII, $65 < 90$
- "235" < "236" car le code numérique lié à '5' est inférieur à celui de '6' (car $53 < 54$)
- "2015" < "3" car le code numérique lié à '2' est inférieur à celui de '3' (car $50 < 51$)

La manipulation des chaînes de caractères nécessite l'utilisation de certaines instructions spécifiques. Ces instructions sont en réalité des algorithmes prédéfinis. On n'écrit pas à nouveau de tels algorithmes, on les utilise directement comme on utilise les fonctions mathématiques prédéfinies ($\ln x$, e^x , $\sin x$, $\cos x$, $\text{abs } x$) en prenant soin de respecter leurs règles d'utilisation. Ces instructions sont des primitives algorithmiques.

On admettra les primitives ci-dessous :

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

longueur (< chaîne >)

Elle fournit la longueur de la chaîne, c'est-à-dire le nombre de caractères situés entre les deux guillemets qui délimitent la chaîne, y compris les espaces.

Exemple : longueur ("YAO Kassi Benoît"), résultat : 16 (car il faut aussi compter les espaces)

code (< caractère >)

Elle fournit la valeur du code numérique lié au caractère.

Exemple : code ('A'), résultat : 65

car (< nombre >)

Elle fournit le caractère correspondant à cette valeur numérique.
Elle est la réciproque de **code** (<caractère >).

Exemple : car (65), résultat : 'A'

+ est la fonction de *concaténation* de chaînes. Elle recopie une chaîne à la fin d'une autre.

Exemple : "Algo" + "rithmique", résultat : "Algorithmique"
"YAO" + ' ' + "Kassi", résultat : "YAO Kassi"

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

Longueur (< chaîne >)

Elle fournit la longueur de la chaîne, c'est-à-dire le nombre de caractères situés entre les deux guillemets qui délimitent la chaîne, y compris les espaces.

☞ **Exemple** : longueur (« GBAME Gbede Sylvain »), résultat : 19 (car il faut aussi compter les espaces)

code (< caractère >)

Elle fournit la valeur du code numérique lié au caractère.

☞ **Exemple** : code ('A'), résultat : 65

car (< nombre >)

Elle fournit le caractère correspondant à cette valeur numérique.

Elle est la réciproque de **code** (<caractère >).

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

☞ **Exemple** : car (65), résultat : 'A'

+ est la fonction de concaténation de chaînes. Elle recopie une chaîne à la fin d'une autre.

☞ **Exemple** : "Algo" + "rithmique", résultat : "Algorithmique"

« GBAME" + ' ' + « Gbede", résultat : « GBAME Gbede"

sschaîne (< chaîne >, < position >, < nombre >)

Elle extrait de < chaîne >, à partir de la position < position >, < nombre > caractères.

☞ **Exemple** : sschaîne("Algorithmique", 5, 7), résultat : "rithmiq"

On extrait de la chaîne "Algorithmique", à partir de la position 5 (donc à partir de la lettre 'r'), 7 caractères.

Chapitre 1 : Variables, types, instructions et expressions

1.3 Les types de variables

cvchaine (< nombre >)

Elle convertit une variable de type numérique, en une variable de type chaîne de caractères.

☞ **Exemple** : cvchaine (7), résultat : "7" ; cvchaine (3.45), résultat : "3.45"

cvnombre (< chaîne de caractères >)

Elle convertit une variable de type chaîne de caractères, contenant une valeur numérique, en une variable de type numérique.

☞ **Exemple** : cvnombre ('7'), résultat : 7

Chapitre 1 : Variables, types, instructions et expressions

1.4 Les Expressions

1. Rappel : l'affectation

L'affectation est une opération qui consiste à stocker une valeur dans une variable ou dans une constante.

Cette opération est représentée par une flèche orientée à gauche : \leftarrow , **et** a pour format général **< identificateur > \leftarrow < valeur >**.

La valeur est soit un nombre, soit une chaîne de caractères, soit une variable du type de l'identificateur, soit une *constante* du type de l'identificateur, soit une **expression** dont le résultat de l'évaluation est du type de l'identificateur.

Chapitre 1 : Variables, types, instructions et expressions

1.4 Les Expressions

2. Définition : L'expression

Une expression est un ensemble de valeurs reliées par un ou plusieurs opérateurs, et qui fournit une seule valeur.

☞ Exemple :

Soient la variable delta (de type à déterminer), et les variables a, b et c de type réel.

Considérons l'instruction d'affectation : **delta** \leftarrow **b*b - 4*a*c**

La partie **b*b - 4*a*c** (située à droite de la flèche) est une **expression**. On veut stocker sa valeur dans la variable delta.

Le calcul de cette valeur est effectué avec les valeurs contenues dans les variables **a**, **b** et **c**.

L'affectation est possible si la variable delta est du même type que le résultat de l'expression **b*b - 4*a*c**. Alors la variable doit être du type réel.

Chapitre 1 : Variables, types, instructions et expressions

1.4 Les Expressions

3. Caractéristiques d'une expression

Une expression possède ***une valeur*** et ***un type***. Une expression peut alors être aussi bien une variable qu'une constante, de n'importe quel type.

Par conséquent, le format général d'une affectation peut s'écrire :
< identificateur > ← < expression >

L'expression peut être :

- un nombre ou une chaîne de caractères,
- une constante du type de l'identificateur,
- une variable du type de l'identificateur,
- un ensemble de valeurs reliées par un ou plusieurs opérateurs dont le résultat de l'évaluation est du type de l'identificateur,

Chapitre 1 : Variables, types, instructions et expressions

1.4 Les Expressions

- une ***fonction*** dont la valeur de retour est du type de l'identificateur. (La notion de fonction sera étudiée dans le chapitre traitant des ***sous-algorithmes***.)

Les notions d'***affectation*** et d'***expression*** sont essentielles en algorithmique

Chapitre 1 : Variables, types, instructions et expressions

1.4 Les Expressions

TRAVAUX DIRIGÉS

Exercice 0 : À chacun son verre, à chacun son jus

On a du jus de gnanmankou dans un verre bleu, et du jus de bissape dans un verre jaune. Proposer un schéma montrant une méthode permettant d'échanger les contenus des deux verres ?

Exercice 0a : Échanger les contenus de deux variables

Écrire les instructions qui permettent d'échanger les valeurs de deux variables.

On supposera que ces deux variables sont a et b de type entier et qu'elles contiennent respectivement les valeurs 10 et -10.

Indication : S'inspirer de la méthode de résolution de À chacun son verre, à chacun son jus.

Chapitre 1 : Variables, types, instructions et expressions

1.4 Les Expressions

TRAVAUX DIRIGÉS

Exercice 0b : Permutation triangulaire

Soient trois variables x , y et z (supposées du même type). Écrivez les instructions permettant de permuter leurs valeurs, de telle sorte que la valeur de x passe dans z , celle de z dans y , et celle de y dans x . On utilisera une seule variable supplémentaire.

Exercice 0c : Une question pour gagner 1500 FCFA

Que trouvera-t-on dans c à l'exécution de l'instruction $c \leftarrow 14$, sachant que c est une constante de type entier contenant la valeur 7 ?

Exercice 1 : Calcul du cube d'un nombre

Écrire un algorithme qui calcule le cube d'un nombre saisi au clavier.

Exercice 2 : Somme, produit et moyenne

Écrire un algorithme qui calcule la somme, le produit et la moyenne de trois nombres saisis au clavier.

Chapitre 1 : Variables, types, instructions et expressions

1.4 Les Expressions

Exercice 3 : Autre expression de la durée

On donne une durée en jours. Écrire un algorithme qui permet d'exprimer cette durée en années, mois, semaines et jours.

(Par exemple : 433 jours équivalent à une durée de 1 an 2 mois 1 semaine et 1 jour)

Exercice 4 : Une petite facture d'achat

Écrire un algorithme qui lit le prix HT en francs CFA et la quantité d'un article, et qui retourne le prix total TTC correspondant, sachant que la TVA vaut 18%. L'algorithme devra afficher une facture détaillée.

Exercice 5 : Extraction d'initiales

Écrire un algorithme qui extrait les initiales d'une personne. On suppose que la personne donne le nom et deux prénoms.

Chapitre 1 : Variables, types, instructions et expressions

1.4 Les Expressions

Exercice 6* : Résultat d'un candidat à un concours

Écrire un algorithme qui affiche le résultat d'un candidat au test d'entrée au cycle d'ingénieur de l'INSTI, selon les données du tableau ci-dessous :

FIN DU COURS