

Université M'Hamed BOUGARA - Boumerdes
Faculté des sciences – Département d'Informatique

Algorithmique et structure de données

Présentée par

Iza lyla

lylaiza@yahoo.fr

Contenu du module

- Introduction
- Historique
- Généralités et concepts de base
- Algorithmes séquentiels simple
- Les structures conditionnelles (en langage algorithmique et en C)
- Les boucles (en langage algorithmique et en C)
- Les tableaux et les chaînes de caractères
- Les types personnalisés

Introduction

L'informatique, est la science du traitement automatique de l'information. Apparue au milieu du 20^{ème} siècle. L'informatique est présente dans tous les domaines de notre vie quotidienne : industrie, gestion, calculs scientifiques et techniques, enseignement, télécommunication, jeux, etc. L'informatique a connu une évolution extrêmement rapide, voici quelques événements importants :

historique

- XIIIe : fabrication de l'Ars Magna, par Raymond Lulle (b) : il s'agit d'une "machine logique" faite de cercles concentriques contenant des mots qui, disposés dans un certain ordre, forment des questions tandis que d'autres mots y répondent
- XVIe : invention du codage binaire par Francis Bacon (a) et du logarithme (à l'origine créé pour simplifier des calculs compliqués) par Napier (b)
- 1624 : Wilhem Schickard construit une "horloge automatique calculante" à Heidelberg
- 1642 : Blaise Pascal, à 19 ans, crée la "Pascaline", machine à calculer mécanique à base de roues dentées, capable de faire des additions et des soustractions (b) ; le langage informatique PASCAL sera plus tard ainsi nommé en son honneur
- 1673 : Leibniz, grand mathématicien, améliore la Pascaline en y ajoutant la multiplication et la division ; par ailleurs, il s'intéresse beaucoup à la numérotation binaire avec laquelle il essaie de concevoir une "caractéristique universelle" dont l'objectif est de réduire toutes les opérations logiques à un calcul (b)

- 1642 : Blaise Pascal, à 19 ans, crée la "Pascaline", machine à calculer mécanique à base de roues dentées, capable de faire des additions et des soustractions (b) ; le langage informatique PASCAL sera plus tard ainsi nommé en son honneur
- 1673 : Leibniz, grand mathématicien, améliore la Pascaline en y ajoutant la multiplication et la division ; par ailleurs, il s'intéresse beaucoup à la numérotation binaire avec laquelle il essaie de concevoir une "caractéristique universelle" dont l'objectif est de réduire toutes les opérations logiques à un calcul (b)
- XVIIIe : La Mettrie, philosophe disciple de Descartes, radicalise la philosophie de ce dernier et écrit *L'homme machine*, où il argumente en faveur d'une vision mécaniste du vivant (c) (Descartes lui-même aurait construit un automate à visage humain). Les automates sont très à la mode à cette époque. L'horloger suisse Vaucansson en construit de très célèbres parmi lesquels un joueur de flûte et un canard pourvu de fonctions locomotrices et digestives, exposés à paris en 1738 : leur fonctionnement utilise un "arbre à came" (comme dans les boîtes à musique), codage binaire du mouvement (a). Un célèbre "joueur d'échec artificiel" parcourt aussi les cours européennes à la fin du siècle (il aurait notamment battu Napoléon) avant qu'on ne démasque la supercherie : un nain caché sous la table actionnait en fait le mécanisme (c).

- 1805 : Jacquart crée les métiers à tisser automatiques, qui utilisent des "programmes" sous forme de cartes perforées, également utilisées dans les pianos mécaniques (a)
- 1818 : Mary Shelley publie "Frankenstein", où l'électricité donne l'étincelle de vie à un être composé à partir de morceaux de cadavres (c)
- 1822 : l'ingénieur anglais Babbage fait les premiers plans de sa "machine à différences", sorte de machine à calculer mécanique utilisant les logarithmes (b) : trop complexe pour la technologie de l'époque, elle ne sera construite d'après ces plans qu'au XXIème siècle.
- 1832 : invention du langage Morse, pour coder les caractères de l'alphabet (c'est un code binaire, composé uniquement des deux symboles : un trait court et un trait long) (a)
- 1833 : Babbage conçoit sa "analytical engine", encore plus performante (et compliquée) que la "machine à différence", utilisant des cartes perforées pour enchaîner l'exécution d'instructions élémentaires sur un calculateur universel (mécanique) : il passera sa vie et se ruinera à essayer en vain de construire sa machine (a, b). Il sera aidé par Lady Ada Lovelace, fille du poète Lord Byron, qui écrira les premiers "programmes" qu'aurait pu exécuter la machine (le langage de programmation ADA sera ainsi nommé pour lui rendre hommage). Cette machine aurait pourtant répondu aux besoins croissants en calcul dans la société anglaise, notamment pour l'astronomie et la navigation.

- 1854 : Le logicien anglais Georges Boole publie son livre *The Mathematical Analysis of Logic*, où il définit les opérateurs logiques dits "booléens", fondés sur deux valeurs 0/1 pour coder Vrai/Faux (a)
- 1876 : Bell invente le téléphone (a)
- 1884 : L'ingénieur américain Hollerith dépose un brevet de machine à calculer automatique
- 1890 : Hollerith commercialise des machines à calculer électriques, utilisées notamment pour traiter automatiquement les données d'un recensement aux Etats-Unis (b). Les besoins industriels en calcul automatique se multiplient.
- 1896 : Hollerith crée une société appelée "Tabulation Machine Corporation", qui deviendra en 1924, "International Business Machine" (IBM), qui existe toujours...
- 1921 : invention du mot "robot" par Karel Capek, auteur dramatique tchèque, dans une de ses pièces (c)
- 1925 : Vannevar Bush, ingénieur américain, construit un calculateur analogique au MIT (Massachusetts Institute of Technology, prestigieuse école d'ingénieur américaine)
- 1927 : la télévision et la radio deviennent opérationnels (a)
- 1931 : l'allemand Konrad Zuse construit un calculateur automatique, le Z1 (b)

- 1936 : Alan Turing propose sa définition des "machines de Turing" et Church invente le "lambda-calcul", qui se révèlent avoir des capacités de calcul équivalentes (b)
- 1938 : fondation de Hewlett Packard, société de matériels électroniques
- 1939 : John Atanasoff et Clifford Berry, son étudiant, conçoivent un prototype appelé ABC à l'université de l'Iowa, reconnu comme le premier ordinateur digital (b)
- 1939-1945 : pendant la guerre,
 - Alan Turing travaille dans le service anglais de décryptage des messages secrets allemands (codés suivant le système appelé "Enigma") : il réalise une machine à décrypter qui contribuera à la victoire des alliés (a, b) ; en 1941, il construit le " Colossus " à l'université de Manchester (bientôt suivi du Mark I et du Mark II), premiers ordinateurs européens avec le Z3 de Konrad Zuse qui, pour la première fois, propose un contrôle automatique de ses opérations
 - John Von Neumann, travaille sur les calculs de balistique nécessaires au projet *Manhattan* (conception et réalisation de la première bombe atomique américaine) (b).
- 1945 : John Von Neumann écrit un rapport où il propose l'architecture interne d'un calculateur universel (ordinateur), appelée désormais "architecture de Von Neumann".

- 1946 : construction de l'ENIAC à l'Université de Pennsylvanie, dernier gros calculateur électrique programmable (mais pas universel) : il fait 30 tonnes, occupe $160m^2$ et sa mémoire est constituée de 18 000 tubes à vide, sa puissance est équivalente à celle d'une petite calculatrice actuelle (b) ; pendant ce temps, Wallace Eckler et John Mauchly conçoivent le Binac (Binary Automatic Computer), qui opère pour la première fois "en temps réel" mais ne sera construit qu'en 1949, avec l'apport de Von Neumann
- 1947 : invention du transistor (qui peut être vu comme un interrupteur miniature)
- 1948 : Claude Shannon publie sa *Théorie mathématique de l'information*, où est introduite la notion de quantité d'information d'un objet et sa mesure en bits (a) ; l'année suivante il construit la première machine à jouer aux échecs

Plan du cours 1: Généralités et concepts de base

- Informatique
- Matériel
- Logiciels
- Programmes
- Langage de programmation
- Compilateurs

1- Informatique

Le terme informatique est proposé en 1962 par Philippe Dreyfus, il désigne le traitement automatique de l'information : il est construit sur la contraction de l'expression « information automatique ». Ce terme a été accepté par l'Académie française en avril 1966, et l'informatique devint alors officiellement la science du traitement automatique de l'information, où l'information est considérée comme le support des connaissances humaines et des communications dans les domaines techniques, économiques et sociaux (figure 1.1). Le mot informatique n'a pas vraiment d'équivalent aux Etats-Unis où l'on parle de Computing Science (science du calcul) alors que Informatics est admis par les Britanniques.

L'informatique traite de deux aspects complémentaires : les programmes immatériels (logiciel, software) qui d'écrivent un traitement à réaliser et les machines (matériel, hardware) qui exécutent ce traitement.

2- Matériel

Le matériel informatique est un ensemble de dispositifs physiques (microprocesseur, mémoire, écran, clavier, disques durs. . .) utilisés pour traiter automatiquement des informations (données). Dans ce contexte, l'ordinateur est un terme générique qui désigne un équipement informatique permettant de traiter des informations selon des séquences d'instructions (les programmes) qui constituent le logiciel.

3- Logiciel

En informatique, un *logiciel* est un ensemble structuré d'instruction interprétables par une machine et d'un jeu de données nécessaires à ces opérations. Le logiciel détermine donc les tâches qui peuvent être effectuées par la machine, en lui procurant son utilité fonctionnelle. Les séquences d'instructions appelées *programmes* et les données du logiciel sont ordinairement structurées en fichiers. La mise en œuvre des instructions du logiciel est appelée exécution et la machine chargée de cette mise en œuvre est appelée ordinateur ou calculateur.

4- Programme

Un programme informatique appelée aussi codage est un ensemble d'opérations destinées à être exécutées par un ordinateurs. Il est une étape importante du développement de logiciels.

Un programme, il est donc un code écrit par un informaticien dans un *langage de programmation*. Il peut être *compiler* (par un logiciel appelé *compilateur*) vers une forme binaire ou directement *interpréter* (par un logiciel nommé *interpréteur*).

Durant l'écriture d'un programme, on peut être confronté à 2 types d'erreur :

- *Les erreurs syntaxiques* : elles sont détectées lors de la compilation et sont le résultat d'une mauvaise écriture du programme.
- *Les erreurs sémantiques* : survient lorsque le programmeur conçoit mal son programme, elles sont liée à la mauvaise analyse du problème. Lors d'une *erreur sémantique*, le programme peut continuer de s'exécuter mais ne fournira pas le résultat souhaité elles sont beaucoup plus graves car elles peuvent se déclencher en cours de l'exploitation du programme.

5- Langage de programmation

Un langage de programmation est un ensemble de mots et de symboles qui servent à exprimer des opérations, des instructions et des structures de données.

Un langage de programmation est un langage informatique, permettant à un humain d'écrire un code source qui sera analysé par un ordinateur. Le code source subit ensuite une transformation ou une évaluation dans une forme exploitable par la machine (fait par un compilateur ou par un interpréteur). Les langages permettent souvent de faire abstraction des mécanismes bas niveaux de la machine, de sorte que le code source représente une solution puisse être rédigé et compris par un humain.

6- Compilateur

Tout langage possède un compilateur ou du moins un **interpréteur**, qui sert à traduire le code source d'un programme écrit dans un langage évolué (exemple c) en code objet exécutable par l'ordinateur. Il permet aussi d'analyser le programme source pour détecter les erreurs de syntaxe commises par le programmeur.

- **N'hésitez pas à poser ou envoyer vos questions**
- **RDV au cours suivant**

Plan du cours 2: Notion de Base sur les Algorithmes

- Introduction
- Structure d'un algorithme
- Partie déclaration
- Partie action
- Construction d'un algorithme
- Traduction en langage de programmation c

Introduction

- L'algorithmique est la science des algorithmes. L'algorithme s'intéresse à l'art de construire des algorithmes ainsi qu'à caractériser leur validité, leur robustesse, leur réutilisabilité, leur complexité et leur efficacité.
- Un algorithme est une suite ordonnée d'instruction, qui indique la démarche à suivre pour résoudre une série de problème, une fois exécutée correctement, conduit à un résultat donné.
- L'algorithme est le résultat de l'analyse d'un problème en utilisant des méthodes, des règles mathématique et de la logique pour que celui-ci soit présenté à une machine (ordinateur).

En d'autre termes, l'algorithme est le résultat de la décomposition d'un problème complexe en opérations élémentaires à exécuter en plusieurs étapes successives par une machine.



Exemple

Pour bien illustrer la notion d'algorithme, prenons l'exemple d'un chauffeur d'ambulance venant récupérer un malade qui s'égare en cours de route; on va lui monter le chemin à suivre pour atteindre son objectif. on suppose qu'on connaît très bien le chemin, alors on lui dira par exemple :

« Suivez cette rue jusqu'au carrefour, puis tournez à droite et à environ 100 mètres prenez à gauche, votre adresse est en fin de rue à droite. »

Les étapes de l'algorithme s'expriment comme suite :

- Suivre la route jusqu'au carrefour
- Prenez à droite
- Roulez environ 100 mètres
- Prenez à gauche
- S'arrêter en fin de rue à droite

Structure d'un algorithme

- Les éléments syntaxique d'un algorithme

L'alphabet	Représente l'ensemble des caractères utilisés dans la langue latine. Exemple : a, b, E,B,S,...
Les mots réservés (mots clés)	C'est des mots qui sont réservés à l'algorithme. Exemple : Algorithme, Var, Debut.....
Les identificateurs	C'est une suite de caractères alphanumérique. Exemple Ecole, test2, ...
Les valeurs	Représente le contenu d'un objet. 15, 7, 'e', faux, ...
Les expressions	C'est une combinaison d'objets et d'opérateurs. On peut avoir des expressions arithmétiques ou logique

La structure d'un algorithme

La structure générale d'un algorithme est :

Algorithme <nom de l'algorithme>;

< Partie déclaration>

Debut

<Partie Actions>

Fin.

Entête de l'algorithme (programme)

Permet d'identifier le nom de l'algorithme, commence toujours par le mot clé : **Algorithme**, cette ligne se termine par un point-virgule

Partie Déclaration

Dans cette partie, on déclare les données nécessaires, pour l'algorithme : les fonctions et procédures ainsi que les variables et constantes.

Partie Actions (ou corps de l'algorithme)

Le corps de l'algorithme ou la partie actions ; délimiter par les mots clé **debut** et **fin**. Cette partie contient les instructions (traitements ou actions) de l'algorithme.

Partie déclaration

La partie déclaration permet de spécifier quelles seront les variables et constantes utilisées au cours de l'algorithme ainsi que le type des valeurs qu'elles doivent respectivement prendre.

1. La *déclaration d'une constante* est réalisée comme suite :

Const < identificateur > = < valeur > ;

2. La *déclaration d'une variable* est réalisée comme suite :

Var < identificateur > : < type > ;

Exemple :

Const a = 2;

Var x : entier;

Partie déclaration (suite)

- **Variables et constantes**

Une *variable* est un espace mémoire nommé, de taille fixe, peut prendre plusieurs valeurs au cours de déroulement d'un algorithme. Ce changement de valeurs se fait par l'opérateur *d'affectation* (\leftarrow). Une *constante* quand a elle ne prend qu'une valeur unique au cours du déroulement de l'algorithme.

- **Identificateurs et mots clés**

Un *identificateur* est une chaîne alphanumérique (composé de caractères et de chiffres), dont le premier caractère est alphabétique pouvant utiliser le tiret de soulignement (_). Il ne doit pas contenir de **caractères spéciaux** (-+*/.?:;?§!...) ni **caractères accentués**.

Un *mot clé* est un mot réservé, il a une utilisation spéciale dans un algorithme ou dans un programme comme par exemple : **Algorithme, Var, Const, if, else, ...**

Partie déclaration (suite)

• Types de données

En algorithmique on trouve deux catégories de type de données, type simple et complexe (structuré). Voici un tableau qui résume les types simples.

Type	description	Syntaxe	Exemple
Entier	La valeur appartient à l'ensemble des entiers relatifs \mathbb{Z} ;	[+ / -] <chiffre>	+25; -12, 123...
Réel	La valeur appartient à l'ensemble des réels \mathbb{R}	[+/-]<partie entière>.<partie fraction> [E<entier >]	+22,52, -12,96, 13,2 E (13,2*12)
Booléen	La valeur 'vrai' ou 'faux'		
Caractère	La valeur prend un caractère de l'ensemble alphabétique {'A','B',...,'z'} ou numérique {'1','2',...} ou caractères spéciaux {'-','+', '<', '....'}, le blanc		'a', '4',

Les actions de base (instructions)

1. Affectation (\leftarrow ou $:=$)

Permet d'affecter une expression à une variable :

$\langle \text{identificateur} \rangle \leftarrow \langle \text{expression} \rangle$

Exemple : $x1 \leftarrow 3$; $y3 \leftarrow x1-7$;

Expressions

Une expression est une combinaison de plusieurs opérandes et opérateur. Il existe plusieurs types d'expressions:

1. **Les expressions arithmétique :** elle font intervenir les operateurs arithmétique (+,-,*,/) ainsi que les deux opérateurs :
 - **DIV :** fourni la partie entière de la division entre deux opérandes. exemple : 5 DIV 2 donne 2
 - **MOD :** fourni le reste de la division entière entre deux opérandes. Exemple : 5 MOD 2 donne 1.

Les actions de base (Expressions)

2. **Les expressions booléennes (logique)** : Elles font intervenir les opérateurs logique (and, or, not,...) ainsi que les opérateurs de relation (<,>,<=,>=, = , <>,...). Exemple $a < b$, $c = a$, des expression logiques.

Priorité des opérateurs

Une expression est évaluée suivant la priorité des opérateurs, on commençant par le plus prioritaire. Dans le cas ou deux opérateurs ont la même priorité, on commence par la plus à gauche.

Les règles de priorité sont les suivantes :

1. les sous expressions entre parenthèses sont évaluées avant d'intervenir dans le reste des calculs en commençant par les plus internes.
2. Unaires : not, -
3. Opérateurs multiplicatifs : * , / .
4. Opérateurs additifs : +, - .
5. Opérateurs relationnels : =, <, >, <=, >=, <>.

Les actions de base (Instructions d'entrées sorties)

Les instruction E/S permettent un échange d'information entre l'algorithme et l'utilisateur. En distingue deux type d'instruction d'E/S :

2. **L'instruction de lecture** : est une instruction qui permet la lecture d'informations ou de données à partir d'un clavier. Sa syntaxe est :

Lire (<liste de variables>);

Exemple : Lire (x); , Lire (x,t);

3. **L'instruction d'écriture** : est une opération de sortie qui permet d'afficher sur l'écran la valeur d'une expression. Sa syntaxe est :

Ecrire (<résultats>);

Exemple : Ecrire (x,y); Ecrire ('le résultat est :', x);

4. **Les commentaires** : est un texte facultatif constitué en générale de phrases, sous mis entre symboles par exemple : '{' et '}'. Les commentaires n'a aucun effet sur le fonctionnement de l'algorithme. Elle sert à expliquer le pourquoi de certaines instructions utilisées dans un algorithmes.

Construction d'un algorithme simple

Pour construire un algorithme on passe par les étapes suivantes :

- Définition du problème
 - Analyse
 - Structuration
 - Traduction
 - Compilation
- Quand il s'agit de la programmation

Pour bien illustrer les étapes prenons l'exemple suivant :

Exemple : Comment calculer la somme de deux nombres avec une calculatrice?

Construction d'un algorithme simple

1. Définition du problème

Ça consiste à avoir exactement ce qu'on attend comme résultat.

- *Application sur exemple :*

- Problème ? => Calculer la somme de 2 nombres

2. Analyse

Dans cette étape il faut identifier toutes les variables nécessaires, ainsi que l'ensemble des actions élémentaires permettant de résoudre ce problème.

- *Application sur exemple :*

Les objets : 2 variables pour stocker le premier et deuxième nombre, une variable pour stocker le résultat. Les actions élémentaires :

- L'utilisateur donne les deux nombres (action composée)
- L'utilisateur donne nombre 1 (opération de lecture)
- L'utilisateur donne nombre 2 (opération de lecture)
- Calcule la somme des 2 nombres (opération de calcul : addition)
- Stocker le résultat dans une variable (affectation)
- Donner le résultat à l'utilisateur (opération d'écriture)

3. Structuration

Identifier la démarche à suivre pour arriver au résultat en partant de l'ensemble d'objets et d'actions et la traduire en pseudo-code algorithme ou organigramme.

Application sur l'exemple précédent :

```
Algorithme somme ;  
    var x, y, reslt : entiers;  
debut  
    ecrire ('donner les valeurs des variables x et y');  
    lire (x,y);  
    reslt  $\leftarrow$  x+y;  
    ecrire ('le resultat de la somme de x et y est :',reslt);  
fin
```

Traduction en langage c

Une fois *l'algorithme* déterminé et le problème solutionné, il faut le traduire dans un *langage de programmation* dans le but de l'utiliser sur la machine. Le *programme* sera alors *compilé* puis exécuté grâce à un compilateur. Il existe plusieurs langages de programmations, dans ce cours nous travaillerons en *langage C*. Le tableau, ci-après, résume la traduction des éléments vu précédemment en algorithmique dans le langage C.

<i>Algorithme nom_algo;</i> déclaration des types et des fonctions déclaration des variables/constantes <i>debut</i> instructions {commentaire} autres_instructions <i>Fin.</i>	déclaration des types et des fonctions <i>int main(void)</i> <i>{</i> déclaration des variables/constantes instructions /* commentaire*/ autres_instructions <i>}</i>
--	--

<p><i>Affectation : ←</i></p> <p><i>arithmétiques : +, -, *, /, div, mod.</i></p> <p><i>logiques : non, et, ou,</i></p> <p><i>comparaison de nombres : =, <>, <, >, <=, >=</i></p>	<p><i>Affectation : =</i></p> <p><i>arithmétiques : +, -, *, /, /, %.</i></p> <p><i>logiques : !, &&, </i></p> <p><i>comparaison de nombres : ==, !=, <, >, <=, >=, =</i></p>
Entier, Reel, Char, Boolean	Int, float ou double, char, bool
<i>lire (a);</i>	<p>ajouter au début avant le <i>main ()</i> :</p> <p>#include <stdio.h></p> <ul style="list-style-type: none"> • si a est un entier/booléen : scanf("%d", &a); • si a est un réel : scanf("%f",&a); • si a est une chaîne : scanf("%s", a);
<p><i>ecrire (a);</i></p> <p><i>ecrire ('la valeur de l'entier a est : 'a);</i></p>	<p>ajouter au début avant le <i>main ()</i> :</p> <p>#include <stdio.h></p> <ul style="list-style-type: none"> • si a est un entier/booléen : printf("%d", a); • si a est un réel : printf("%f",a); • si a est une chaîne : printf("%s", a); <p>printf("La valeur de l'entier est %d", a);</p>

<pre> <i>si</i> (condition) <i>alors</i> <instructions_1> <i>sinon</i> < instructions_2> <i>finsi</i> </pre> <p>NB : le bloc sinon est optionnel en algorithmique et obligatoire en c</p>	<pre> If (condition) { <instruction1>} else { <instruction2>} </pre>
<p>Déclaration de variables :</p> <p>Var x : entier</p> <p>Var x, y, z : entier;</p>	<pre> type <nom_var> ; </pre> <p>Exemple : Int x ;</p> <p>On peut déclarer plusieurs variables de même type au même temps. Par exemple : <i>int</i> x,y,z ;</p>

Exemple 2 : écrire un algorithme qui calcule la différence et le produit de deux nombres

```
Algorithme prod_diff ;  
var x, y, prod, diff : entieres;  
debut  
    ecrire ('donner les valeurs de x et y');  
    lire (x,y);  
    prod  $\leftarrow$  x*y;  
    diff  $\leftarrow$  x-y ;  
    ecrire ('le produit des deux variables x et y est :', prod);  
    ecrire ('la difference des deux variables x et y est :', diff);  
fin
```


Traduction de l'exemple 2 en c :

```
#include <stdio.h>
Void main(){
    int x,y , prod,diff;
    printf ("donner les valeurs de x et y \n" );
    scanf( "%d,%d" &x,&y);
    prod = x*y;
    diff = x-y;
    printf (" la valeur du produit est %d \n ",prod);
    printf ("la valeur de la difference de x et y est: %d \n " ,
diff);
}
```

- **N'hésitez pas à poser ou envoyer vos questions**
- **RDV au cours suivant**