

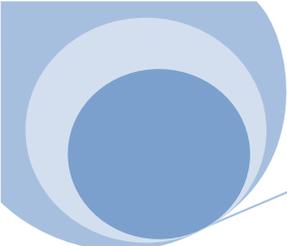
Informatique

Département pharmacie Oran

Année 2008/2009

Faculté de médecine Oran





L'initiation en informatique :

I- Définition du terme informatique :

L'informatique est la science du traitement de l'information par une machine automatique (ordinateur). L'informatique vient de la contraction des mots information, automatique. Il a été proposé en 1962 par Philippe Dreyfus et accepté par l'Académie française en 1966. L'informatique est d'une importance capitale en science dans l'industrie et dans l'administration.

II- L'information sa forme et son sens :

L'information est un élément de connaissance pouvant être communiqué, en distinguant dans une information la forme et le sens.

La forme : le code.

Le sens : l'interprétation de l'information

III- Ordinateur :

Néologisme proposé en 1956 par Jacques Perret à la demande du constructeur IBM, c'est une machine électronique programmable de traitement de l'information numérique, on parle ainsi de Hardware, software.

Hardware : pour désigner l'ensemble des éléments matériels de l'ordinateur

Software : pour désigner la partie logicielle

IV- Architecture de l'ordinateur :

Un ordinateur se compose essentiellement d'une unité centrale et des unités d'entrée-sortie (d'E/S), en plus des mémoires secondaires indispensables pour une exploitation informatique efficace.

IV.1- Unité centrale :

C'est le cerveau de l'ordinateur. Sur le plan fonctionnel, il est composé d'une unité de traitement et d'une mémoire centrale reliées par un bus de données.

IV.1.1- Unité de traitement (processeur) :

Elle exécute automatiquement le programme instruction par instruction. Elle se divise en deux parties :

- ALU (Arithmétique Logique Unité)

Effectue les opérations arithmétiques de base (comparaison, addition, soustraction, multiplication, la division...)

- CU (Contrôle Unité)

Responsable des contrôles nécessaires pour synchroniser les différentes tâches des autres composants.

IV.1.2- Mémoire centrale :

Elle est directement accessible par l'unité de traitement (le processeur) et contient le code du programme (ou partie de lui) en cours d'exécution ainsi que les données qui lui sont nécessaires. La mémoire centrale est souvent dénommée vive ou RAM (Random Access Memory) c'est-à-dire une mémoire permettant la lecture et l'écriture par contre la mémoire morte ou ROM (Read Only Memory) et une petite mémoire réservée uniquement à la lecture.

IV.2- Unité d'entrée-sortie :

Elle permet la communication avec l'homme ou avec d'autres machines telles que le clavier, l'écran, la souris, l'imprimante, le scanner

IV.3- Mémoires auxiliaires :

Elle permet le stockage des données et des programmes, les mémoires auxiliaires sont la plus grande partie du temps réalisées sur des supports magnétiques comme la disquette mais de nos jours des supports optiques comme les disques compacts.

La communication avec l'unité centrale est assurée par le bus d'une unité d'E/S (lecteur)

V- Traitement de l'information :

Le traitement de l'information par une machine automatique consiste à faire élaborer par cette machine des informations appelé résultat a partir d'information connue appelé donné.

Les systèmes de numérotation :

Un système de numérotation se définit par deux éléments : le premier est la base et le deuxième c'est les symboles.

En informatique les systèmes les plus utilisés sont les suivants :

Système	Base	Les symboles
Décimale	10	0, 1, 2, 3, 4, 5, 6, 7, 8, 9.
Binaire	2	0, 1.
Octale	8	0, 1, 2, 3, 4, 5, 6, 7.
Hexadécimale	16	0 → 9, A, B, C, D, F.

Soit N un nombre quelconque exprimé dans une base B le nombre N sera noté comme suit :

$$N = (a_{n-1} a_{n-2} a_{n-3} \dots a_0)_B$$

Tel que B est la base du système de numérotation, les a_i symboles du système $i = \{0, 1, \dots, (n-1)\}$ avec $a_i < B$

Ex : $N = (19017)_B$

I- Le système binaire :

C'est la base utilisée en informatique pour la représentation des informations aux niveaux machine, ce système possède 2 chiffres (0 et 1), c'est 2 états sont les seuls que la machine peut assimiler

II- Le système décimale :

C'est le système usuel dans la vie quotidienne, la base du système décimal est 10, c'est les symboles sont les chiffres : 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

Cela veut dire qu'en décimal, les 10 chiffres précédents sont suffisants pour exprimer n'importe quel nombre seulement, la machine ne peut assimiler que les 2 valeurs 0 et 1, il serait important de savoir comment exprimer les nombres décimaux en binaire et vice versa, et on parle de conversion de base.

III- Le système octal :

La base du système octal est 8, en octal les nombres sont représentés sous forme de combinaison de chiffres parmi les suivants : 0, 1, 2, 3, 4, 5, 6, 7.

II- Le système hexadécimal :

Le système hexadécimal (base 16) utilise 16 chiffres pour la représentation des nombres à savoir les chiffres du système décimal de 0 → 9 et les 6 premières lettres de l'alphabet, le tableau suivant donne l'équivalence décimale et hexadécimale.

Hexadécimal	Décimal
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	10
B	11
C	12
D	13
E	14
F	15

IV- Passage de la base 2 à la base 10 :

L'exemple suivant illustre la méthode de conversion en décimal d'un nombre exprimé dans une base 10 ou une Base B quelconque :

Soit $N = (a_{n-1} a_{n-2} a_{n-3} \dots a_0)_B$

Pour avoir la représentation en décimal du nombre N exprimé dans une base B quelconque il suffit d'effectuer le calcul suivant :

$$N_{10} = a_{n-1} * B^{n-1} + a_{n-2} * B^{n-2} + \dots + a_0 * B^0.$$

La formule générale s'écrit comme suit :

$$(N)_{10} = \sum_{i=0}^{n-1} a_i B^i$$

Ex : $N = (10111001)_2 = (2^7 + 2^5 + 2^4 + 2^3 + 2^0)_{10} = (185)_{10}$

V- Passage de la base 10 à la base 2 :

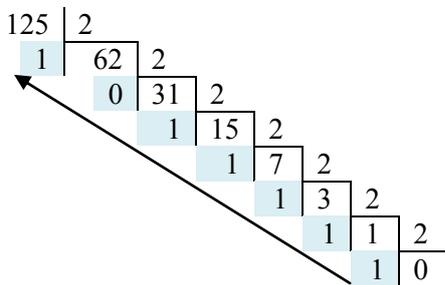
Pour exprimer n binaire un nombre exprimer dans une base B en dispose d'une méthode par division successive.

Conversion par division successive :

Soit x un nombre exprimer dans la base 10, pour exprimer dans une base X il suffit d'effectuer des divisions successives sur B jusqu'à l'obtention d'un résultat nul.

Ex :

$X = (125)_{10} = (1111101)_2$



Passage de la base 8 à la base 10 :

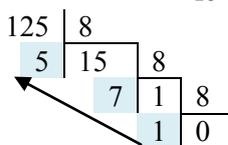
La base 10 étant la base du système usuel, il est intéressant de savoir convertir un nombre exprimé en octale en un nombre exprimer en décimale. Les règles de passage de la base 8 a la base 10 sont les même que pour la base 2.

Ex : $X = (175)_8 = 1 * 8^2 + 7 * 8^1 + 5 * 8^0 = (125)_{10}$

Passage de la base 10 à la base 8 :

Ils s'effectuent dans la même manière que le passage de la base 10 à la base 2

Ex : $X = (125)_{10} = (175)_8$



Passage de la base 16 à la base 10 :

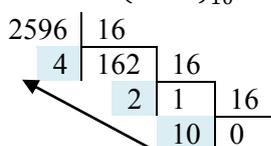
La même formule est utilisée pour le passage des bases 2 et 8 à la base 10

Ex : $X = (A24)_H = 10 * 16^2 + 2 * 16^1 + 4 * 16^0 = (2596)_{10}$

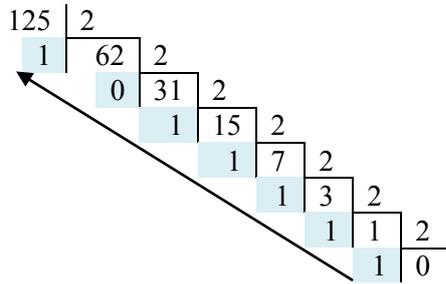
Passage de la base 10 à la base 16 :

La encor il est question d'appliquer la méthode des divisions successives sur la base 16.

Ex : $X = (2596)_{10} = (A24)_{16}$



Exercice : Soit le nombre $(175)_8$ quelle est son résultat en base 2
 $X = (175)_8 = 1 * 8^2 + 7 * 8^1 + 5 * 8^0 = (125)_{10} = (1111101)_2$



Remarque : Lorsqu'une base est une puissance d'une autre base le passage de l'une à l'autre devient très facile et ne nécessite pas une base intermédiaire, ainsi le passage de la base 8 2^3 ou 16 2^4 peut s'effectuer sans passer par la base 10

Passage de l'octale au binaire :

Base 8 est une puissance de la base 2 (2^3). Pour convertir un nombre octale en binaire en procédant comme suit : on a $8 = 2^3$ cela veut dire que pour représenter un seul chiffre en binaire il faut utiliser 3 bits ainsi la représentation des chiffres de la base 8 en binaire :

Chiffres octale	Chiffre binaire
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Ex : $X = (175)_8 = (1111101)_2$
 $\begin{matrix} 1 & 7 & 5 \\ 001 & 111 & 10 \\ (1|111|101)_2 & = & (175)_8 \end{matrix}$

Passage du hexadécimale au binaire :

La base 16 est une puissance de la base 2 (2^4). Pour convertir un nombre hexadécimale en binaire en procédant comme suit : on a $16 = 2^4$ cela veut dire que pour représenter un seul chiffre en binaire il faut utiliser 4 bits ainsi la représentation des chiffres de la base 16 en binaire :

Ex : $X = (A24)_H = (101000100100)_2$

$\begin{matrix} A & 2 & 4 \\ 1010 & 0010 & 0100 \end{matrix}$

$X = (1010|0010|0100)_2 = (A24)_H$

Le hexadécimale permet de remplacer les écritures binaire par des termes 4 fois plus courtes.

Chiffres hexadécimale	Chiffre binaire
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

Notion d'algorithme

I- Définition :

Un algorithme est l'ensemble des étapes ordonné qui permet de résoudre un problème donné. Ex : résolution d'une équation de second degré, dans la vie courante on effectue des algorithmes pour réaliser des actions quotidiennes ex : recettes de cuisine.

II- L'environnement d'un algorithme :

C'est l'ensemble des objets nécessaire utilisé dans l'algorithme par ex : les objets a, b, c, et x dans l'équation $ax^2 + bx + c$ constituent les éléments de l'algorithme pour résoudre cette équation, avant de décrire une resette de cuisine on commence par lister les ingrédients nécessaire a sa réalisation

III- Le programme :

Pour pouvoir exécuté un programme par l'ordinateur il faut le traduire dans un langage de programmation. La solution traduit dans un langage de programmation donné constitue le programme. Chaque langage de programmation possède :



Etape de résolution d'un problème :

La première étape consiste dans l'analyse du problème, le résultat de cette étape est la décomposition du problème en composant élémentaire.

La seconde étape est l'établissement d'un algorithme.

La troisième étape est la traduction de l'algorithme en programme e utilisant un langage de programmation choisi par ex : pascal, Delphi, PHP, html

IV- Caractéristique d'un objet :

On déclare un objet en définissant ses caractéristiques.

Ex : une déclaration de naissance à l'état civile comporte les éléments qui caractérisent le nouveau né à savoir son nom ; prénom

En informatique les caractéristiques générales d'un objet sont :

- L'identificateur (le nom de l'objet)
- Le type de l'objet
- Valeur de l'objet
- Nature de l'objet.

IV.1. L'identificateur d'un objet :

Tout objet est caractériser par un nom ou identificateur qui lui est propre et que l'on utilise pour le désigne sans ambiguïté. Un identificateur est un nom qui respect une syntaxe particulière, il est constitué d'une suite de caractère alphabétique est numérique sauf l'espace est qui doit commencer par un caractère alphabétique, il peut aussi contenir le caractère de soulignement, ex : on peut attribuer a un objet le nom « rayon » ou « valeur absolue » mais non pas « 1x »

Remarque :

Il est préférable pour facilité la lisibilité de l'algorithme d'utilisé des noms significatif.

IV.2. Le type d'un objet :

Un type définie l'ensemble des valeurs que peut prendre un objet ex : entier, réel ... etc.

IV.3. La valeur d'un objet :

La valeur d'un objet est un élément quelconque du domaine de définition décrit par le type.

IV.4. La nature d'un objet :

La nature d'un objet est soit constante ou variable suivant l'algorithme

Nom	Nature	Type	Valeur
Pi	Constante		3.14
R	Variable	Réel	
X	Variable	entier	

V- Structure générale d'un algorithme :

Un algorithme peut être considéré comme une action composée et formalisée comme suit :

Algorithme « nom_alg » ;

« Partie déclaration » ;

Début

« Partie de traitement (corps) : description des actions »

Fin

Fin de l'algorithme « nom_alg »

Ex :

Algorithme Max ;

Var A, B, Max1 : réel ;

Début

Lire (A) ;

Lire (B) ;

Si A>B alors Max1 = A

 Si non Max1 = B

Fin Si

Ecrire (Max1) ;

Fin

Fin de Algorithme Max

Les types élémentaires (expressions est instruction élémentaire)

I- les types élémentaires :

I.1 Les types standards :

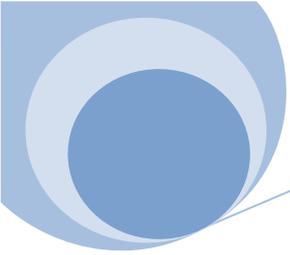
I.1.1 Types logique booléen :

Les 2 constantes de ce type sont noté en représentation externe par vrais et aux et en machine la valeur faux est représentée par 0 et la valeur vrais par 1

Identificateur	Nature	Type	Valeur
B ₁	Variable	Booléen	
True	Constante		Vrai
False	Constante		Faux

I.1.2 Types entier :

La représentation externe d'une constante de type entier est conforme a la syntaxe habituelle en notation décimale (base 10).



Identificateur	Nature	Type	Valeur
E ₁	Variable	Entier	
X ₁	constante		40

1.1.3. Type réel :

Le type réel prend ses valeur dans un sous ensemble des réelles, en représentation externes l'écriture des constantes réelles respect la syntaxe habituelle en notation décimale en utilise le point comme mark décimale

Identificateur	Nature	Type	Valeur
R ₁	Variable	Réel	
Pi	constante		3.14159

1.1.4. Type caractère :

L'ensemble des types caractères est jeu fini et totalement ordonné de caractères (code ASCII) il comporte les lettres, les chiffres, les signes de ponctuation, les symboles utilisé en temps que opérateur +, -, *, <, = Les caractère spéciaux ex %, @, \$; & ainsi des caractères non imprimable dit caractères de contrôles tel que l'espace en représentation externe un constante caractère est représentée par un caractère encadré par deux apostrophes à l'exception de l'apostrophe elle-même qui est doublet, cette syntaxe permet de distinguer une constante caractère d'un identificateur et d'une constante numérique.

Identificateur	Nature	Type	Valeur
X	Variable	Caractère	
A	Constante		'a'

Exemples :

- 5 : est une constante de type entier.
- '5' : est une constante de type caractère.
- x : est un identificateur d'une variable.
- 'x' : est une constante de type caractère.
- 15,7 : est une constante de type réel.
- Faux : est une constante de type booléen

1.1.5. Type chaine de caractère :

Une chaine de caractères est une suite du code ASCII en représentation externe est encadré par 2 apostrophes et l'apostrophes figurent dans une chaine est doublet.

Identificateur	Nature	Type	Valeur
Y	Constante		'13457'
Z	Constante		'samedi'
K	Constante		'Aujourd'hui'

I.2 Les types définis par énumération :

Un type ainsi défini est dit énumérer les constantes de ce type des constantes nommées.

Identificateur	Nature	Type
Jour	Type	(samedi, dimanche, lundi, mardi, mercredi, jeudi, vendredi)
Hier, demain	Variable	Jour

I.3 Types intervalle :

C'est une partie d'un ensemble de valeurs ordonnées déjà définies désignées par borne inférieure borne supérieure.

Identificateur	Nature	Type
Chiffre	Type	0.. 9
C ₁ C ₂	Variable	Chiffre

II- les expressions :

Une expression décrit la façon dont une valeur est obtenue, une expression peut être :

- Soit une constante par exemple : 27, 'Algérie', faux.
- Soit une variable par exemple : B, X ...
- Soit appelée de fonction ex : sin (x)
- Soit une expression complexe qui contient plusieurs opérateurs ex : $a + \frac{b}{5} - abs(y) \times 3$ il faut respecter la priorité des opérateurs dont l'ordre est :
 1. Opérateur unaire : - 5, + 4
 2. Opérateur multiplicatif : /, div., mod, * et opérateur logique.
 3. Opérateur aditif : + et - et les opérateurs logiques.
 4. Opérateur de relation : <, >, ≥, ≤, =, ≠ ...
 5. On cas d'égalité : de gauche à droite.

Les instructions élémentaires

Les instructions élémentaires sont les opérations primitives ou bien de traitement ou sont les objets, c'est l'affectation, la structure et l'écriture.

1. L'affectation :

Permet d'affecter une valeur à une variable. Sa forme est <identificateur> ← <expression>

Ex : x ← 5

y ← 2 * a

Remarque : la valeur affectée doit être dans le même type que la variable ex : x entier ~~x ← 5,5~~

La variable prend la valeur de la dernière affectation attribuée.

Incrémentation et décrémentation :

N ← N+1

N ← N-1

2. La lecture :

Lire une valeur c'est mètre par l'utilisateur une valeur dans une case mémoire a travers le clavier, la valeur attribué doit être de même type que la variable

Sa forme est lire (<liste d'identificateur séparé par des virgules>)

Ex : lire (x), lire (x, y, z)

3. L'écriture :

L'écriture permet d'afficher sur écran le contenu des cases mémoires, sa forme est écrire (<liste des variables séparer par des virgules>)

Ex : Ecrire (A) ; Ecrire (B) ; Ecrire (A, B) ; Ecrire (50) ; Ecrire ('Djamel') ; Ecrire (a*b/2)

Exercice 1 :

Ecrire l'algorithme qui permet de lire 3 nombre réels A, B, C et les visualiser.

Algorithme affichage ;

Var A, B, C : réels ;

Début

Lire (A, B, C) ;

Ecrire (A, B, C) ;

Fin

Fin de l'algorithme affichage ;

Exercice 2 :

Ecrire l'algorithme qui permet de lire 3 nombre réels A, B, C et les afficher dans l'ordre inverse de leur lecture.

Algorithme affichage ;

Var A, B, C : réels ;

Début

Lire (A, B, C) ;

Ecrire (C, B, A) ;

Fin

Fin de l'algorithme affichage ;

Exercice 3 :

Ecrire l'algorithme qui permet de lire le coté d'un carré, de calculer sont périmètre et sa surface et les visualisé.

Algorithme carré ;

Var coté, périmètre, surface : réels ;

Début

Lire (coté) ;

Périmètre $\leftarrow 4 * \text{coté}$;

Surface $\leftarrow \text{coté}^2$;

Ecrire (périmètre, surface) ;

Fin

Fin de l'algorithme carré ;

Exercice 4 :

Ecrire l'algorithme qui permet de lire la largeur et la longueur d'un rectangle et de calculer sont périmètre er sa surface et les visualisé.

Algorithme rectangle ;

Var largeur, longueur, périmètre, surface : réels ;

Début

Lire (largeur, longueur) ;

Périmètre $\leftarrow 2 * (\text{largeur} + \text{longueur})$;

Surface $\leftarrow \text{largeur} * \text{longueur}$;

Ecrire (périmètre, surface) ;

Fin

Fin de l'algorithme rectangle ;

Exercice 5 :

Ecrire l'algorithme qui permet de lire le rayon d'un cercle et de calculer son périmètre et sa surface et les visualiser.

Algorithme cercle ;

Var Rayon, Pi, périmètre, surface : réels ;

Début

Lire (Rayon) ;

Pi ← 3,14 ;

Périmètre ← $R * 2 * Pi$;

Surface ← $R^2 * Pi$;

Ecrire (périmètre, surface) ;

Fin

Fin de l'algorithme cercle;

Les structures de contrôle

On appelle structure de contrôle toutes structures permettant d'effectuer une action (un groupe d'actions) selon une condition.

- Les structures conditionnelles ou alternatives
- Structures répétitives ou interactives.

1. Traitement séquentielle :

C'est une suite d'instruction qui s'exécute l'une à la suite de l'autre

Début

Séquence 1

Séquence 2

.....

Séquence n

Fin

2. Structures conditionnelle :

2.1. Forme alternative simple :

Si (<condition>) alors <actions> ;

Fin Si ;

Ex : Si ($\Delta < 0$) alors écrire ('pas de solution') ;

Fin Si ;

2.2. Forme alternative complète :

Si (<condition>) alors <actions> ;

Sinon <action2> ;

Fin Si ;

Ex : Si (wilaya=16) alors écrire ('Alger') ;

Sinon écrire ('autre ville') ;

Fin Si ;

3. Structure répétitive

3.1 La boucle « Tant que » :

Tant que (<condition>) faire <action> ;

Fin Tant que ;

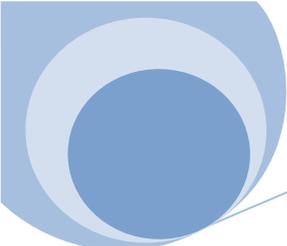
Ex : Tant que (feu n'est pas vert) faire patienter ;

Fin Tant que ;

3.2. La boucle répéter :

Répéter <action> jusqu'à <condition> ;

Fin Répéter ;



Exercice 3 :

Ecrire l'algorithme qui permet de faire la permutation de deux nombre réel

Algorithme permutation

Var x, y, z : réels ;

Début

Lire (x, y) ;

z \leftarrow x ;

x \leftarrow y ;

y \leftarrow z ;

Ecrire (x, y) ;

Fin

Fin de l'algorithme permutation ;

Exercice 4 :

Ecrire l'algorithme qui permet de lire le nom et le montant d'une feuille de maladie d'un passion et d'afficher le message malade doit passer le contrôle lorsque le mentent est strictement supérieur a 1500 DA, dans le cas contrarie, le message sera A rembourser.

Algorithme Maladie ;

Var montant : réel ;

Nom, message : chaine de caractère ;

Début

Lire (nom, montant) ;

Si montant > 1500 alors message \leftarrow 'Malade doit passer le contrôle'

Sinon message \leftarrow 'A rembourser' ;

Fin si ;

Ecrire (nom, message) ;

Fin

Fin de l'algorithme Maladie ;

Exercice 5 :

Ecrire l'algorithme qui permet de calculer la somme d'une suite de 100 nombre donné

Algorithme somme ;

Var S, nb, I : entier

Début

Pour I := 1 à 100 faire

Lire (nb) ;

S \leftarrow S+nb ;

Fin pour ;

Ecrire (S) ;

Fin

Fin de l'algorithme somme ;