الجمهورية الجزائرية الديمقراطيـــة الشعبية

République Algérienne Démocratique et Populaire

وزارة التكوين و التعليم المهنيين

Ministère de la Formation et de l'enseignement Professionnelle

معهد التكوين و التعليم المهنيين – سيدي بلعباس -

Institut de la formation et de l'enseignement Professionnels

- SIDI BELABBES -

MANUEL TECHNIQUE ET PEDAGOGIQUE

INFORMATIQUE

OPTION

BASES DE DONNEES

MODULE

Elaborer des Algorithmes simples

NIV : V

2012

I.F.E.P

WWW.DZPEN.COM | قلم الجزائر

EQUIPE DE PRODUCTION

Rédaction :

M^me ABIB KHADIDJA PSEP 2 en informatique.

Production :

Institut de Formation et Enseignement Professionnels

SANHADRI ABDELHAFID - Sidi Bel Abbes ALGERIE

Validation : (Membres de la commission de validation)

Nom	Prénom	Fonction

Remerciements :

Le concepteur remercie toutes les Personnes qui ont contribué, de près ou de loin à la conception et à la réalisation de ce manuel professionnel jusqu'à l'édition finale.

Elle remercie également les entreprises et organismes qui l'ont gracieusement autorisée à utiliser à des fins pédagogiques les illustrations et les textes tirés de leur documentation.

Les droits de reproduction d'adaptation ou de traduction de ce manuel professionnel sont réservés à l'IFEP, y compris la reproduction par procédé mécanique ou électronique.

WWW.DZPEN.COM | قلم الجزائر

La table des matières

	Parcours général de formation de la spécialité	6
	Objectif du module	7
	Introduction	8
Chapitre1	**Décomposition d'une action algorithmique**	
	1.1 Description de la résolution d'un problème.	11
	1.1.1 Exemple introductif	11
	1.1.2 Définition du problème	12
	1.2 Notion et définition d'un Algorithme.	13
	1.2.1 Exemple introductif	13
	1.2.2 Définition d'un Algorithme	14
	1.2.3 Qualités d'un Algorithme	14
	1.2.4 Structure de base d'un formalisme	15
	1.2.5 Exercice	15
	1.3 Résumé	15
	1.4 Exercice de synthèse	16
Chapitre 2	**Déclaration des types élémentaires des données**	
	2.1 Déclarations	17
	2.1.1 Définition	17
	2.1.2 Exercice	18
	2.2 Types Constantes ; Variables.	18
	2.2.1 Les constantes	18
	2.2.2 Les variables	19
	2.2.2.1 Nom d'une variable	19
	2.2.2.2 Type d'une variable	20

WWW.DZPEN.COM | قلم الجزائر

	2.2.3 Exercice	21
	2.3 Types standards : Entier, Réel, Caractère, Booléen	21
	2.3.1 Type entier	21
	2.3.2 Type réel	21
	2.3.3 Type caractère	22
	2.3.4 Type booléen	22
	2.4 Résumé	23
	2.5 Exercice de synthèse	23
Chapitre 3	**Représentation d'un algorithme**	
	3.1 Enchaînement d'actions.	24
	3.1.1 Définition	24
	3.1.2 Exercice	25
	3.2 Affectation.	25
	3.2.1 La notion d'affection	25
	3.2.2 Fonctionnement de l'instruction d'affection	26
	3.2.3 Exercice	27
	3.3 Lecture et écriture	27
	3.3.1 L'instruction de lecture	28
	3.3.2 L'instruction d'écriture	29
	3.3.3 Exercice	30
	3.4 Commentaires	30
	3.4.1 Définition	30
	3.4.2 Fonctionnement	31
	3.5 Résumé	32
	3.6 Exercice de synthèse	32
Chapitre 4	**Ecriture d'un algorithme qui utilise les structures de contrôle**	
	4.1 Définition des structures de contrôles	33
	4.1.1 Définition	33

WWW.DZPEN.COM | قلم الجزائر

	4.2 Le choix	33
	4.2.1 Alternative	33
	4.2.1.1 La structure alternative **« Si ….alors »**	33
	4.2.1.2 La structure alternative **« Si …. Alors … Sinon »**	34
	4.2.2 Choix multiple	39
	4.2.2.1 La structure alternative **« Cas où »**	39
	4.2.3 Exercice	40
	4.3 La répétition	41
	4.3.1 Répétition contrôlée par des conditions	42
	4.3.1.1 La boucle « Tant que »	42
	4.3.1.2 La boucle « répéter … jusqu'à »	44
	4.3.2 Répétitions avec indice	46
	4.3.2.1 La boucle « Pour »	46
	4.3.3 Exercice	48
	4.4 Résumé	48
	4.5 Exercice de synthèse	49
Chapitre 5	**Ecriture d'un algorithme qui utilise des chaînes de caractères**	
	5.1 Déclaration	50
	5.1.1 Définition et déclaration	50
	5.1.2 Exercice	51
	5.2 Opérations sur les chaînes de caractères	51
	5.2.1 La lecture et l'écriture	51
	5.2.2 L'affectation	52
	5.2.3 La comparaison	52
	5.2.4 La concaténation	53
	5.2.5 Les fonctions et les procédures portant sur les chaînes	53
	5.2.5.1 La fonction « Length »	53
	5.2.5.2 La fonction d'extraction de sous–chaînes « COPY »	54
	5.2.5.3 La fonction de localisation de sous–chaînes « POS »	56

WWW.DZPEN.COM | قلم الجزائر

	5.2.5.4 La procédure de suppression de sous–chaîne dans une chaîne « DELETE »	57
	5.2.5.5 La procédure de conversion d'une chaîne en numérique « VAL »	57
	5.2.5.6 La procédure de conversion du numérique en chaîne « STR »	58
	5.2.5.7 La fonction insérer des caractères dans une chaîne « INSERT »	59
	5.2.6 Exercice	59
	5.3 Résumé	60
	5.4 Exercice de synthèse	60
Chapitre 6	**Ecriture d'un algorithme qui utilise des tableaux à une dimension**	
	6.1 Déclaration d'un tableau : Vecteur	63
	6.1.1 Définition et déclaration	63
	6.1.2 Exercice	64
	6.2 Manipulation d'un tableau : Lecture, écriture	64
	6.2.1 La lecture d'un tableau	64
	6.2.2 L'écriture d'un tableau	67
	6.2.3 Exercice	67
	6.3 Méthodes de tri d'un vecteur	67
	6.3.1 Tri croissant	67
	6.3.2 Tri par sélection	70
	6.3.3 Exercice	71
	6.4 Résumé	71
	6.5 Exercice de synthèse	72
Chapitre 7	**Ecriture d'un algorithme qui utilise des tableaux à deux dimensions**	
	7.1 Déclaration d'une matrice	73
	7.1.1 Définition et déclaration	73
	7.1.2 Exercice	75
	7.2 Manipulation d'une matrice : Lecture, écriture	76
	7.2.1 La lecture d'une matrice	76
	7.2.2 L'écriture d'une matrice	78
	7.2.3 Exercice	78
	7.3 Résumé	79
	7.4 Exercice de synthèse	79

WWW.DZPEN.COM | قلم الجزائر

Chapitre 8	Elaboration des algorithmes simples	
	8.1 Exercice de synthèse global	80
	8.2 Résumé général	80
	8.3 Conclusion générale	82
	8.4 Le corrigé des exercices	83
	8.5 Bibliographie	98

WWW.DZPEN.COM | قلم الجزائر

Parcours général de formation de la spécialité

Le programme d'étude de **2448** heures en « **Informatique / Option : BASES DE DONEES** » est constitué de **20** Modules dont dix (**10**) modules qualifiants « MQ » et dix (10) modules complémentaires « MC », visant chacun une compétence du métier. La durée de formation s'étale sur quatre (04) semestres + 06 Mois en stage pratique en milieu professionnel (Entreprise).

Pour réaliser ces modules, une marche à suivre est proposée. Elle est illustrée par la liste qui suit. Afin de repérage, on a entouré d'un rectangle à filet double, le module « **Elaborer des algorithmes simples** » que vous avez en main.

Code	Désignation des modules	Durée
MQ11	Elaborer des algorithmes simples	136 H
MQ12	Elaborer des algorithmes complexes	68 H
MQ13	Maîtriser un langage procédural.	136 H
MQ14	Ecriture des programmes à l'aide d'un langage procédural.	102 H
MQ15	Finaliser les programmes	68 H
MC11	Structure machine	68 H
M C12	Fichier	68 H
MC13	Utiliser les logiciels de bureautique et exploiter Internet	204 H
MC1.4	Probabilités et Statistiques	68 H
MC1.5	Organisation et gestion des entreprises	34 H

WWW.DZPEN.COM | قلم الجزائر

Objectif du Module

Module : **Elaborer des algorithmes simples**

Code : MQ1.1 Durée : 136 h

OBECTIF MODULAIRE

COMPORTEMENT ATTENDU :

A l'issue de ce module, le stagiaire doit être capable d'élaborer des algorithmes simples.

CONDITIONS D'EVALUATION :

- ✓ ***A partir de :***
  - ➢ Documentation
- ✓ ***A l'aide de*** :

.

CRITERES DE PERFORMANCE :

- ➢ Identification correcte des données et des résultats.
- ➢ Respect des règles d'élaboration d'un algorithme
- ➢ Algorithme optimisé (temps d'exécution le moins possible, le plus petit nombre de variables, moins de code)
- ➢ Déroulement correct de l'algorithme.

WWW.DZPEN.COM | قلم الجزائر

Introduction

Elaborer des algorithmes simples est un module très important dans la spécialité informatique car en le maitrisant vous pouvez comprendre un problème, et décrire la suite des actions élémentaires permettant d'obtenir à partir des données fournies, les résultats escomptés.

Cette description doit être précise, envisager le moindre détail et prévoir les diverses possibilités de données, car le rôle d'un algorithme est fondamental. En effet, sans algorithme, il n'y aurait pas de programme qui n'est que la traduction de l'algorithme dans un langage compréhensible par l'ordinateur.

Au cours de votre apprentissage du module « ***élaborer des algorithmes simples*** » va vous aider à créer vos propres algorithmes en utilisant les techniques d'élaboration des algorithmes tels la séquence, l'affectation, les structures de contrôle , les répétitions et les tableaux pour justement résoudre le problème dans n'importe quels domaine d'application.

Ainsi, ce manuel doit vous guider, à travers un exposé théorique abondamment illustré d'exemples et d'exercices, vers une meilleure perception des problèmes que pose la conception d'un programme fiable, en vous donnant tous moyens de les résoudre.

WWW.DZPEN.COM | قلم الجزائر

Chapitre 1 :

Décomposition d'une action algorithmique

Certains voient, à tort, dans l'ordinateur une machine pensante et intelligente, capable de résoudre bien des problèmes. En fait, celui-ci serait capable de rien si quelqu'un (le programmeur en l'occurrence) ne lui avait fourni la liste des actions à exécuter. Cette description doit être faite de manière non ambigüe car il ne faut pas s'attendre à la moindre interprétation des ordres fournis. Ils seront exécutés de manière purement mécanique.

De plus, les opérations élémentaires que peut exécuter un ordinateur sont en nombre restreint et doivent être communiquées de façon précise dans un langage qu'il comprendra.

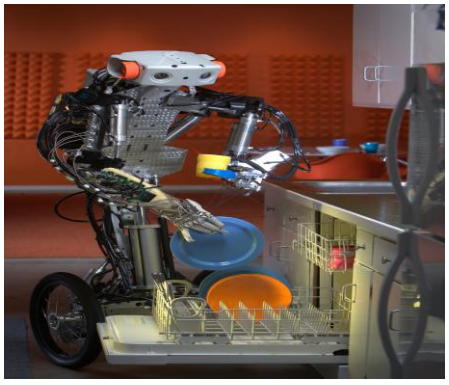
Le problème principal de l'utilisateur est donc de lui décrire la suite d'actions élémentaires permettant d'obtenir, à partir des données fournies, les résultats escomptés. Cette description doit être précise, envisager le moindre détail et prévoir les diverses possibilités de données.

1.1 Description de la résolution d'un problème

Le procédé de résolution d'un problème avec un ordinateur consiste à définir les différentes étapes logiques et ordonnées pour résoudre un problème.

1.1.1 Exemple introductif

Imaginant un robot domestique à qui nous devons lui indiquer les étapes les plus élémentaires lui permettant de préparer une tasse de café soluble.



Une première version des étapes logiques et ordonnées pourrait être :

(1) Faire bouillir de l'eau

(2) Mettre le café dans la tasse

(3) Ajouter l'eau dans la tasse

Les étapes ne sont probablement pas assez détaillées pour que le robot puisse les interpréter. Chaque étape doit être affinée en une suite d'étapes plus élémentaires. Ainsi l'étape

WWW.DZPEN.COM | قلم الجزائر

(1) Faire bouillir de l'eau peut être affinée en

(1.1) Remplir la bouilloire d'eau

(1.2) Brancher la bouilloire sur le secteur

(1.3) Attendre l'ébullition

(1.4) Débrancher la bouilloire

De même,

(2) Mettre le café dans la tasse peut être affinée en

(2.1) Ouvrir le pot à café

(2.2) Prendre une cuiller à café

(2.3) Plonger la cuiller dans le pot

(2.4) Verser le contenu de la cuiller dans la tasse

(2.5) Fermer le pot à café

Et **(3) Ajouter l'eau dans la tasse** pourrait être affinée en :

(3.1) Ajouter de l'eau dans la tasse jusqu'à ce que celle-ci soit pleine

1.1.2 Définition du problème

Définir un problème consiste à formaliser le plus précisément possible ce qu'il faut accomplir par écrit.

Il y'a trois étapes qui caractérisent la résolution d'un problème sur ordinateur :

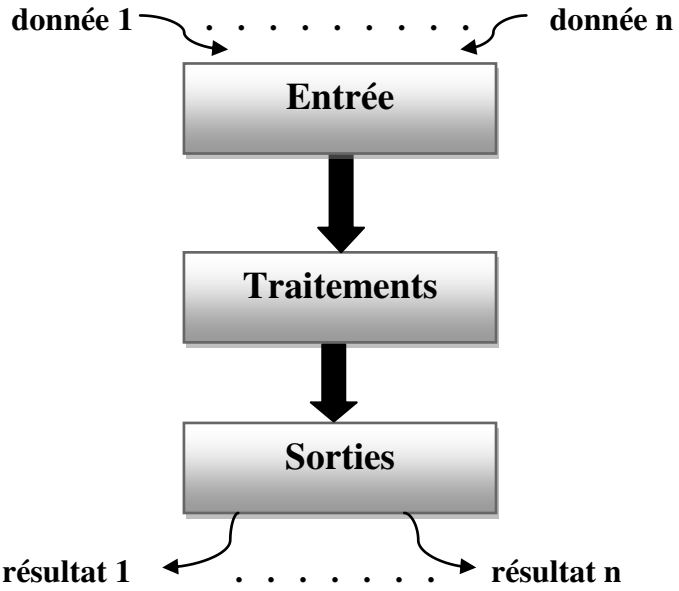
- **Comprendre la nature du problème** posé et préciser les données fournies en entrées.
- **Préciser les résultats** que l'on désire obtenir en sorties.
- **Déterminer le processus de transformation** des données en résultats.

WWW.DZPEN.COM | قلم الجزائر

Ces trois étapes ne sont pas indépendantes et leur ordre peut être modifié.

Donc pour tous problème afin qu'il puisse être automatisés il faut spécifier trois éléments :

- Les données à traités.
- Les traitements à effectuer.
- Les résultats à obtenir.



1.2 Notion et définition d'un Algorithme

1.2.1 Exemple introductif :



Selon vous que fait un stagiaire chaque matin au réveil ?

Voyons ensemble un exemple des étapes qu'effectuera un stagiaire le matin :

1. Le réveil sonne
2. Je me lève
3. Je me lave
4. Je m'habille
5. Je prépare le café
6. Je déjeune

WWW.DZPEN.COM | قلم الجزائر

7. Je prends mes affaires

8. J'ouvre la porte

10. Je sors

Nous pouvant constater sur cet exemple simple que l'ordre des opérations a de l'importance. En effet, on ne peut pas intervertir les actions 2 et 3 ou encore 6 et 7 et ainsi de suite.

1.2.2 Définition d'un algorithme

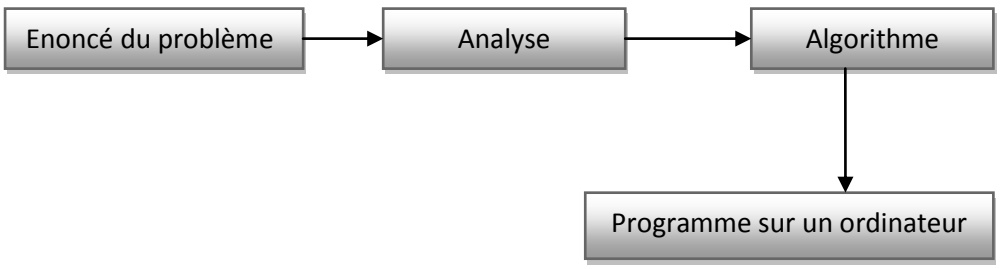
Un algorithme est une suite finie de règles à appliquer dans un ordre déterminé à un nombre fini de données pour arriver, en un nombre fini d'étapes, à un certain résultat, et cela indépendamment des données.

De façon plus précise, un algorithme est la description d'un traitement destiné à être réalisé sur un ordinateur.

1.2.3 Qualités d'un algorithme

Tout programme fourni à l'ordinateur n'est que la traduction dans un langage de programmation d'un algorithme mis au point pour résoudre un problème donné.

Algorithmique



Programmation

Pour obtenir un bon programme, il faut partir d'un bon algorithme. Il doit, entre autres, posséder les qualités suivantes :

- Etre **clair,** facile à comprendre par tous ceux qui le lisent

WWW.DZPEN.COM | قلم الجزائر

- Présenter la plus **grande généralité** possible pour répondre au plus grand nombre de cas possibles

- Etre d'**une utilisation aisée** même par ceux qui ne l'on pas écrit et ce grâce aux **messages** à l'écran qui indiqueront quelles sont les données à fournir et sous quelle forme elles doivent être introduites ainsi que les différentes actions attendues de la part de l'utilisateur

- Etre conçu de **manière à limiter** le nombre d'opérations à effectuer et la place occupée en mémoire

- **Décomposer le problème** en plusieurs sous-problèmes c'est-à-dire **affinements** des différentes étapes.

1.2.4 Structure de base d'un Algorithme

Un algorithme se compose :

1) D'un en-tête précisant :

- Le nom donné à l'algorithme ;
- La déclaration des données de l'algorithme ;

2) D'un corps formé d'une suite d'ordres ordonnés.

3) Les résultats fournis par l'algorithme.

```
Algorithme    nom de l'algorithme
Déclaration
        Donnée 1, . . . , Donnée n

Debut
        action 1
                .
                .
        action n

    résultat 1, . . . , résultat n

Fin
```

1.2.5 Exercice :

Prenons l'exemple d'une classe de stagiaires qui passe l'examen d'algorithmique. L'enseignant corrige et informe ses stagiaires de leurs notes. Quelles sont les étapes élémentaires pour calculer la moyenne de tous les stagiaires.

1.3 Résume :

En terminant ce premier chapitre sur la décomposition d'une action algorithmique, vous êtes maintenant en mesure d'analyser un problème et à déterminer ses données, les résultats auxquels on doit aboutir et les opérations élémentaires qui mènent à ces résultats.

Les données se trouvent, ainsi, transformées d'étape en étape jusqu'à atteindre les résultats. C'est à ce niveau que votre créativité est mise à profit.

Voici un résumé des principaux éléments vus au cours de ce chapitre.

WWW.DZPEN.COM | قلم الجزائر

La **première section** vous a permis de comprendre que pour résoudre un problème donné il faudrait l'analyser et le formaliser pour qu'il puisse être automatisé pour cela vous devez le définir et le comprendre, déterminer les données fournies et les résultats attendus et enfin trouver comment le résoudre.

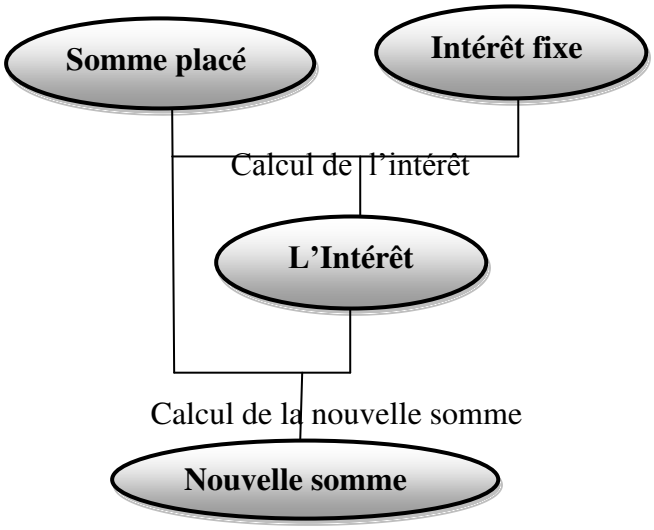
La **deuxième section** vous a donné la définition d'un algorithme qui est une suite d'opérations élémentaires devant être exécutées dans un ordre donné, pour accomplir une tâche donnée. Vous avez vu aussi comment rendre un algorithme clair et compréhensible. Et de considérer un problème dans son ensemble, à préciser les données fournies et les résultats à obtenir puis à décomposer le problème en plusieurs sous-problèmes plus simples.

Toutes ces notions sont essentielles à la compréhension d'un problème et à son analyse et par la suite son développement.

1.4 Exercice de synthèse :

Dans une banque on propose aux clients qui placent une **somme d'argents**, un **taux d'intérêt fixe** pendant 1 an.

Pouvez-vous indiquer les étapes logiques et élémentaires pour calculer **l'intérêt** et la **nouvelle somme d'argents**.



WWW.DZPEN.COM | قلم الجزائر

Chapitre 2 :

Déclaration des types élémentaires des données

WWW.DZPEN.COM | قلم الجزائر

Un très grand nombre de traitements peuvent être automatisés, donc gérés à l'aide d'un ordinateur. Il faut pour cela décider d'une représentation concrète à l'aide d'objets manipulables par les algorithmes qui décrivent l'enchaînement des opérations. Ensembles d'objets d'une part, description d'actions d'autre part, telles sont les deux composantes d'un algorithme, sur lesquelles ce chapitre et les suivants fournissent toutes les explications nécessaires.

2.1 Déclaration

2.1.1 Définition

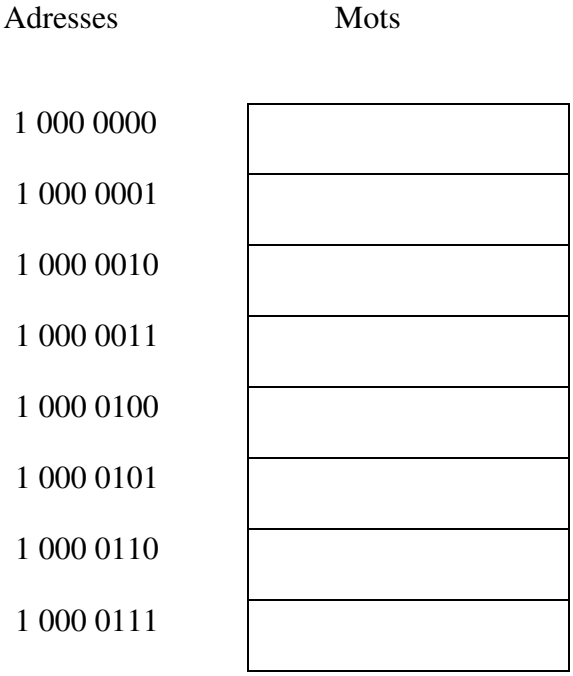


Figure 1 : La mémoire centrale de l'ordinateur

La mémoire centrale de l'ordinateur est formée d'éléments dont chacun ne peut prendre que deux états distincts. Ce sont les bits d'information.

C'est pour cette raison que chaque mot de la mémoire est repéré par un numéro qui constitue **son adresse**.

Dans un langage de programmation, les adresses de la mémoire centrale sont représentées par des noms. **(on appellera plus tard des variables)**

Le programmeur ne connaît que les noms X, Total, Prix, ….. Il ne se préoccupe pas des adresses qui leurs seront allouées en mémoire. Il y'a donc deux façons de voir la mémoire centrale de l'ordinateur : côté programmeur et côté ordinateur.

WWW.DZPEN.COM | قلم الجزائر

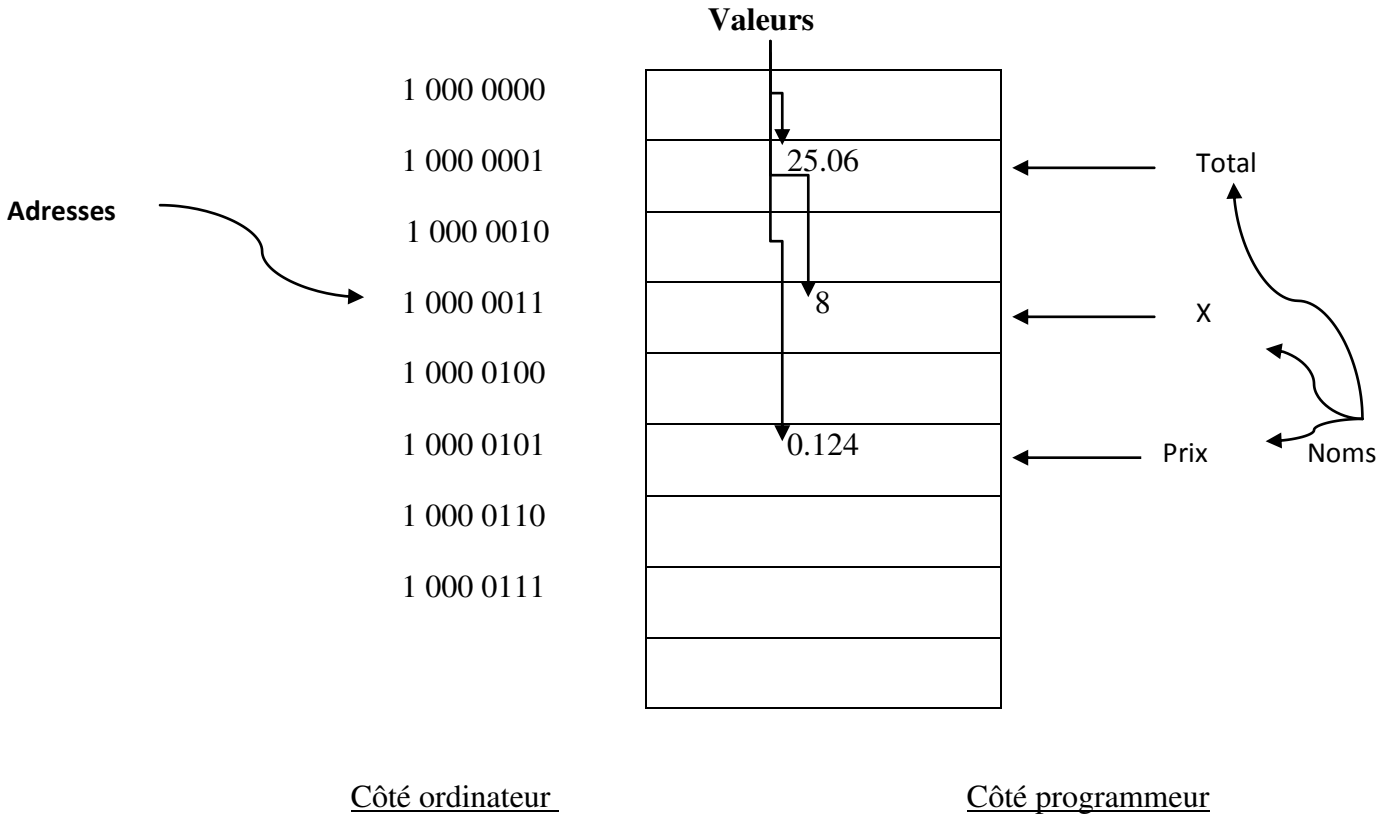


Figure 2 : Deux façons de voir la mémoire centrale.

2.1.2 Exercice

1. Comment sont représentés les mots dans la mémoire centrale ?

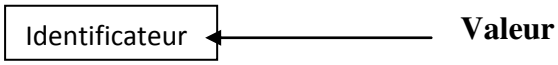
---

2. Quelles sont les deux façons de voir la mémoire centrale ?

2.2 Type Constantes, Variables

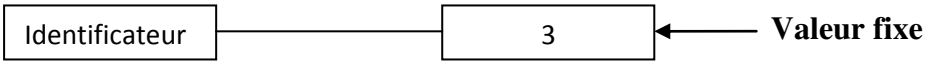
2.2.1 Les constantes

une valeur numérique ou une chaine de caractère est associé à **un nom** de donnée ou **un identificateur.**



WWW.DZPEN.COM | قلم الجزائر

On appelle un identificateur une constante si on lui attribue l'élément de donnée (valeur) de manière définitif c'est-à-dire l'élément ne change pas tout au long du traitement.



- Une constante doit toujours être définie avant de figurer dans un traitement à l'intérieur d'un algorithme.
- La forme générale de la déclaration d'une constance est :

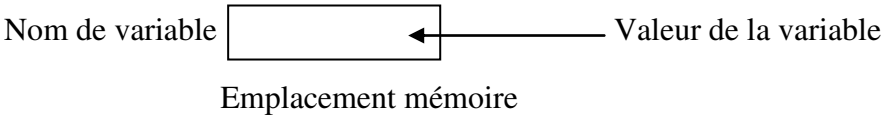
Constante identificateur = valeur

Exemple :

Constante PI = 3.14

2.2.2 Les variables

On a vu précédemment qu'une adresse mémoire correspond à un nom, ce nom est appelé en programmation, une **variable.**



2.2.2.1 Nom d'une variable

Le choix des noms des variables, on appelle aussi identificateur de la variable, est très important. En effet, choisir des noms représentatifs des informations qu'ils désignent permet d'avoir des programmes lisibles.

Exemple :

Dans une facture, utiliser l'identificateur de la variable **Prix** est un meilleur choix que X pour désigner le prix d'un produit.

Important :

D'une manière générale, dans tous les langages de programmation, un nom de variable est formé d'une ou plusieurs lettres (sans espace, en un seul mot).

Les chiffres sont également autorisés, à condition de ne pas apparaître au début du nom. Par contre, le nombre maximum de caractères autorisés varie d'un langage à un autre.

WWW.DZPEN.COM | قلم الجزائر

Exemples :

Rayon, surface, Pi sont des noms corrects.

Prix ttc, 27xyz sont des noms incorrects car le 1er contient un espace et le second commence par un chiffre.

2.2.2.2 Type d'une variable

L'ordinateur ne peut traiter qu'une information exprimée sous forme binaire. Or, celle-ci peut être de nature différente (par exemple des nombres et des lettres). Par conséquent, pour distinguer un genre d'information d'un autre, il faut employer plusieurs codes différents.

La connaissance du code binaire ne suffit pas pour déterminer l'information correspondante. Il est nécessaire de savoir comment la valeur qui s'y trouve a été codée. Cette distinction correspond à la notion de **type**.

Ainsi, une variable destinée à recevoir des nombres est de type « numérique », et une variable destinée à contenir des lettres sera dite de type « caractère ».

Pour cela, vous utiliserez des instructions particulières nommées *déclarations de type*.

La déclaration de variables de manière générale est comme suit :

```
Variable
        Identificateur : type de variable
```

Exemple :

Variable

Prix, S, Total : numérique

Signifie que les variables nommées Prix, S et Total sont de type numérique.

Variable

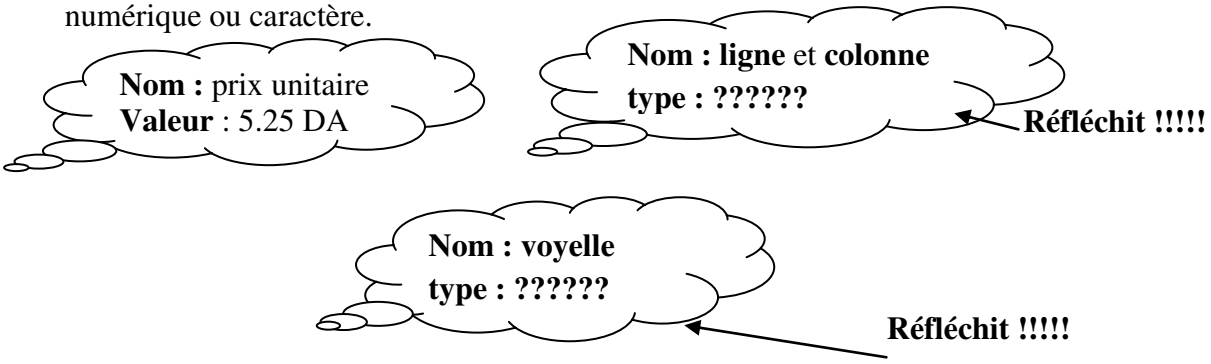
Lettre, C : caractère

Signifie que les variables nommées Lettre et C sont de type caractère.

WWW.DZPEN.COM | قلم الجزائر

2.2.3 Exercice

A travers cet exercice vous allez vous exercer à déclarer des variables de type numérique ou caractère.



2.3 Types standards : Entier, Réel, Caractère, Booléen

2.3.1 Type entier

Il s'agit du type des nombres qui sont des entiers relatifs.

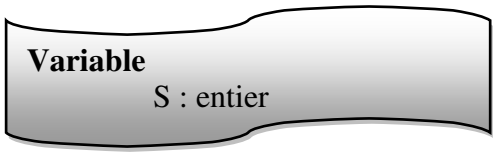
Un nombre entier ne contient pas de point décimal, ni un exposé.

Il se compose simplement d'une suite de chiffres précéder ou non d'un signe (+, -).

Exemple : 3, 18 , +54 , -625 , ……

Mais pas 3.5 ou 'A'.

Déclaration :



2.3.2 Type réel

Un nombre réel doit contenir un point décimal ou un exposé (ou les deux).

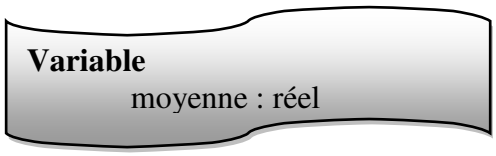
Si le nombre réel contient un point décimal celui-ci doit figurer entre deux chiffres.

Exemple : 4 , 3.5 , 45.74 e -1 , 0.12

Mais pas 'A'.

WWW.DZPEN.COM | قلم الجزائر

Déclaration :



2.3.3 Type caractère

Il s'agit du type qui contient les caractères.

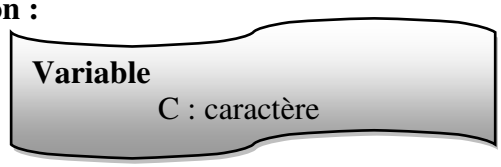
Les caractères peuvent être minuscules ou majuscules.

Les chiffres peuvent être des caractères, il n'est pas question de faire des opérations dessus !

Exemple : 'A' , 'b' , '7'

Mais pas 'AA'.

Déclaration :



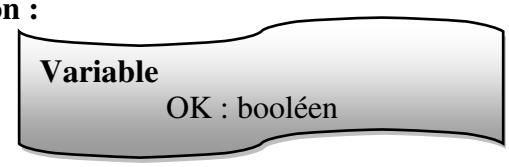
2.3.4 Type booléen

Les variables du type booléen ne peuvent prendre que deux valeurs,

TRUE (vrai) ou **FALSE** (faux).

Il n'est possible de les demander directement à l'utilisateur. Par contre, il est toujours possible de les écrire.

Déclaration :



WWW.DZPEN.COM | قلم الجزائر

2.4 Résumé

Vous venez de voir dans ce chapitre les données et leurs structures. Plus précisément, dans la **première section** qui était consacrée à la représentation en mémoire des données : chaque mot de la mémoire est repéré par un numéro qui constitue son adresse. Et les adresses de la mémoire centrale sont représentées par des noms.

La deuxième section vous a proposé les constantes et les variables.

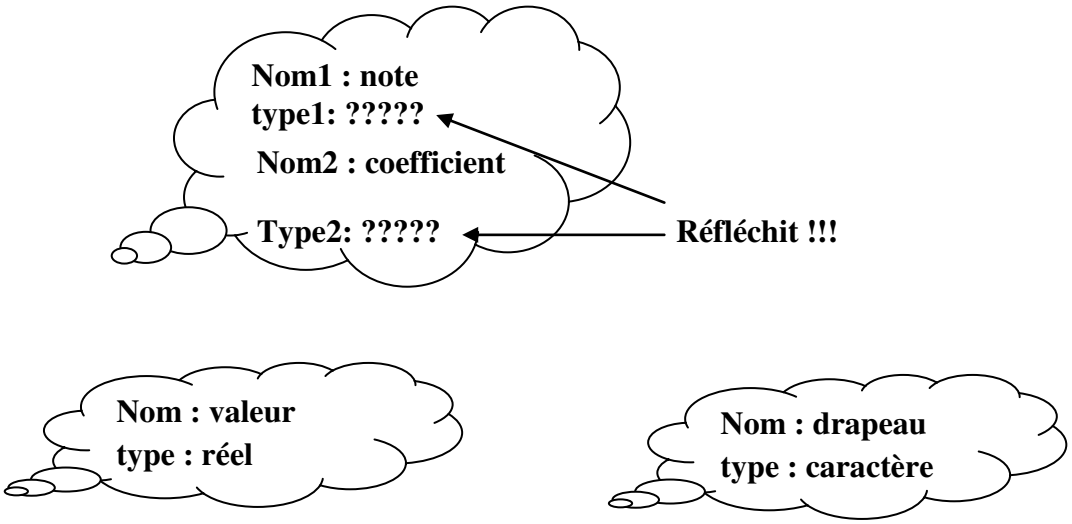
Une constante est un identificateur on lui attribue une valeur de manière définitive avant de figurer dans un algorithme.

Une variable est un identificateur dont la valeur peut changer au cours de l'exécution. Pour chaque variable if faut spécifier le nom et le type.

Enfin**, la troisième section** vous a permis de connaître les différents types standards : le type Entier, le type Réel, le type Caractère, le type Booléen.

2.5 Exercice de synthèse

A travers cet exercice vous allez vous exercer à déclarer des variables de type différent.



WWW.DZPEN.COM | قلم الجزائر

Chapitre 3 :

Représentation d'un algorithme

Un algorithme décrit des actions, enchaînées selon un ordre rigoureux défini et validé par l'analyse. Ce chapitre présente des structures séquentielles, actions exécutées dans l'ordre de leur apparition dans l'algorithme.

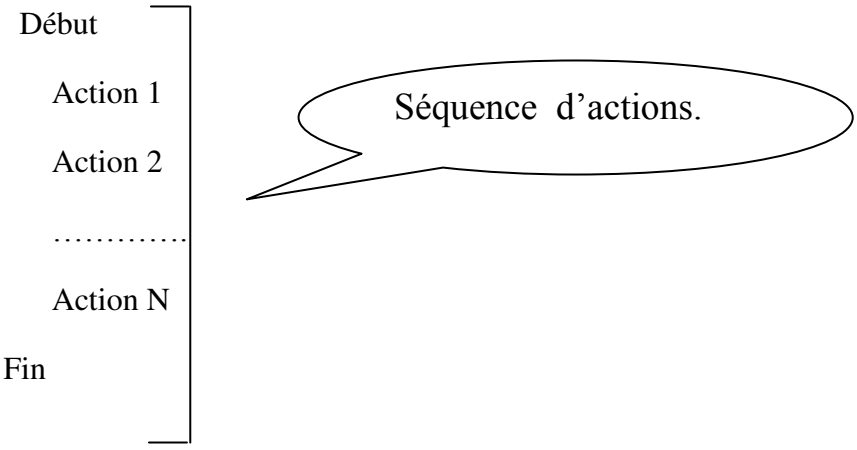
Dans ce chapitre on présentera d'abord l'affectation ensuite la lecture et l'écriture et on terminera avec les commentaires.

3.1 Enchaînement d'actions

3.1.1 Définition

Une action dans un algorithme est un ordre qui déclenche l'exécution d'un traitement.

Enchaînement d'actions c'est l'exécution de l'une à la suite de l'autre.



Exemple :

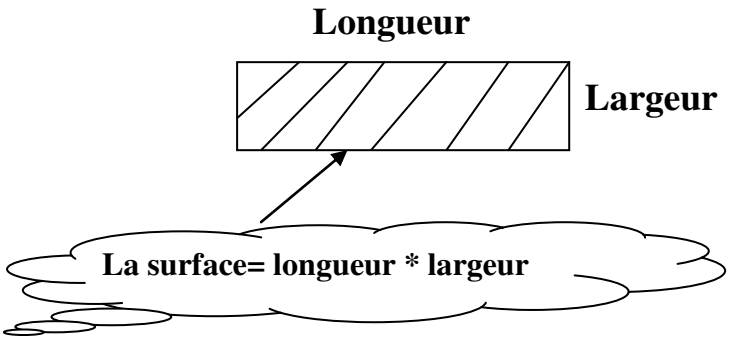
Soit à additionner deux nombres A et B, les opérations à effectuer se présentent ainsi :

1. Entrer la valeur de A
2. Entrer la valeur de B
3. Additionner A et B (A+ b)
4. Afficher le résultat.

Les opérations de 1 à 4 s'exécutent séquentiellement, c'est-à-dire l'une à la suite de l'autre.

WWW.DZPEN.COM | قلم الجزائر

3.1.2 Exercice



Problème : calculer la surface d'un rectangle.

On vous aidant du schéma et de la formule trouvez l'enchaînement des opérations pour résoudre le problème.

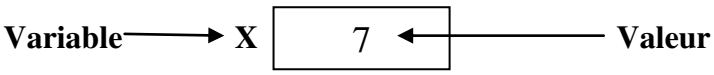
3.2 Affectation

Dans le chapitre précédent, nous avons vu ce que sont les variables. Dans cette section, nous allons étudier l'une des actions qui permettent de manipuler leurs valeurs, à savoir l'instruction d'affectation.

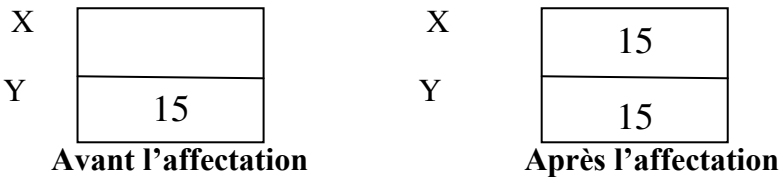
3.2.1 La notion d'affection

L'affectation (ou assignation) est une instruction simple qui consiste à placer une valeur dans une variable.

1. Une instruction d'affectation va permettre, par exemple, **d'affecter à la variable X la valeur 7**. On peut dire également ranger dans X la valeur 7.



2. La valeur à ranger dans une variable pourra également provenir d'une autre variable. Dans ce cas, on ira par exemple, **affecter à X la valeur de Y.**



3. D'une façon générale, on pourra affecter à une variable le résultat d'un calcul numérique.

WWW.DZPEN.COM | قلم الجزائر

Par exemple, ranger dans **Z** le résultat de **X + Y**.

Z	30

Dans ce chapitre et dans les suivants, nous conviendrons d'employer le symbole ←

Ainsi, l'instruction « affecter à la variable X la valeur 7 » s'écrira donc : **X ← 7**

De même :

X ← Y Signifiera affecter à X la valeur de Y, ou bien : X reçoit la valeur de Y.

3.2.2 Fonctionnement de l'instruction d'affection

De façon générale, l'instruction d'affectation s'écrit :

<Variable> ← <expression>

Où **<variable>** désigne le nom d'une variable destinée à recevoir une valeur, et **<expression>** désigne la valeur en question.

L'instruction d'affectation se déroule en deux étapes :

Elle détermine d'abord la valeur de l'expression située à droite du symbole ← , puis elle range le résultat dans la variable située à gauche.

Exemple 1 :

(1) **X ← 2**

(2) **Y ← X+1**

(3) **X ← Y+4**

	X	Y
(1)	2	----
(2)	2	3
(3)	7	3

Vous remarquez que :

- Lorsque l'on détermine la valeur d'une expression, cela ne modifie pas la valeur des variables qui apparaissent dans cette expression. Ainsi, Y ← X+1 détermine la valeur de Y sans modifier celle de X ;
- Lorsque la variable réceptrice comporte déjà une valeur, celle-ci est remplacée par celle qui vient d'être déterminée, comme dans X ← Y + 4.

WWW.DZPEN.COM | قلم الجزائر

Exemple 2 :

En procédant de la même façon que ci-dessus, et en exécutant les instructions dans l'ordre où elles sont écrites, nous obtenons :

	A	B	C
A⟵ 2	**2**	**---**	**---**
B⟵ 3	2	**3**	**---**
C ⟵A+B	2	3	**5**
⟵C	**5**	3	5
C⟵ B-1	5	3	**2**
B⟵A+C-1	5	**6**	2

3.2.3 Exercice

X | **15** | **Y** | **10**

On obtient :

10	**15**

- Donner les instructions d'affectation qui permettent d'échanger les valeurs de deux variables (c'est –à- dire la permutation entre deux variables).
- Ensuite exécuter les instructions obtenues dans un tableau comme dans l'exemple ci-dessus.

3.3 Lecture et Ecriture

Pour qu'un programme ait un intérêt pratique, il doit communiquer à son utilisateur un certain nombre d'informations ou de résultats, par l'intermédiaire d'un périphérique, par exemple l'écran. Ces résultats sont communiqués grâce à l'instruction d'écriture.

D'autre part, dans certains cas, l'utilisateur d'un programme peut être amené à lui transmettre des informations ou des données, également par l'intermédiaire d'un périphérique, par exemple le clavier. Ces données sont transmises grâce à

WWW.DZPEN.COM | قلم الجزائر

l'instruction de lecture.

3.3.1 L'instruction de lecture

L'instruction de lecture va nous permettre de fournir des valeurs à notre programme.

De façon plus précise, l'instruction de **lecture** va recevoir une valeur sur un périphérique (le clavier, par exemple) et va l'attribuer à une variable.

Supposons que votre algorithme doit recevoir de son utilisateur, lors de son exécution, la valeur d'une variable. Si cette variable est nommée.

Par exemple Prix, vous devez écrire l'instruction suivante :



Exemple :

Considérons le programme suivant :

Variables X, Y, P : **entier**

Debut

X ← 5

Y ← 7

P ← X*Y

Fin

Remarque :

Le symbole * signifie multiplier. Il est utilisé à la place du symbole x dans la plupart des langages de programmation pour ne pas le confondre avec la variable X.

Si vous exécutez le programme ci-dessus plusieurs fois, il produira toujours le même résultat, à savoir : 5 7 35

Pour que ce programme ait un intérêt, il faut qu'il soit plus général, c'est-à-dire qu'il soit capable de fournir le résultat de la multiplication de deux nombres quelconques. Dans ce cas, le programme s'écrira comme suit :

WWW.DZPEN.COM | قلم الجزائر

Variables X, Y, P : **entier**

Debut **Valeurs saisies par l'utilisateur**

Lire X ,Y

P ← X*Y

Fin

Les valeurs de X et Y ne sont plus constantes maintenant, mais elles sont saisies par l'utilisateur lors de l'exécution du programme.

3.3.2 L'instruction d'écriture

Cette instruction a pour rôle de fournir à d'un programme des informations ou des résultats.

Supposons, par exemple, que votre programme effectue le calcul du montant d'une facture. Une fois le calcul terminé, vous aimeriez peut-être visualiser ce montant. Dans ce cas, il suffit de donner l'ordre à votre programme de vous **écrire** ce résultat à l'écran. Si le total de la facture a été affecté à une variable que vous avez nommé par exemple montant, cet ordre s'écrira alors comme suit :

écrire montant

De la même manière, vous pouvez visualiser le contenu de plusieurs variables en même temps :

écrire X, Y , total

Exemple :

Soit l'algorithme suivant :

Variables A, B, C : **entier**

Debut

A ← 9

B ← 12

C ← A + B

WWW.DZPEN.COM | قلم الجزائر

écrire A, B, C

Fin

A son exécution, l'algorithme produira les résultats suivants :

9	12	21

3.3.3 Exercice

Variables A, B, C, D : **entier**

Debut

Lire A, B

C ← A + B

D ← A * B

écrire C, D

Fin

On va attribuer respectivement 6 , -5 à A et B. Que sera les valeurs de C et D ?

A	**6**	**B**	**-5**

C	**?**	**D**	**?**

3.4 Commentaire

3.4.1 Définition

Variables A, B, C : **entier**

Debut

Lire A, B

C ← A * B

écrire C

Fin

WWW.DZPEN.COM | قلم الجزائر

Lorsqu'on exécute l'algorithme ci-dessus, on lui fournit deux données, par exemple 9 et 10, et les résultats se présentent comme suit :

9	10	90

Comme vous pouvez le voir, il n'y a aucune explication qui accompagne les résultats. On pourrait souhaiter obtenir quelque chose de plus explicite du genre :

Nombre saisis : 9		10
Produit :	90	

Ou bien : **9 fois 10 égal 90**, ou tout simplement : **9 x 10 = 90**

3.4.2 Fonctionnement

Ceci est possible dans tous les langages de programmation. Les textes qui accompagnent les résultats s'appellent des **commentaires.** Pour les obtenir, il suffit de faire apparaître ces commentaires dans la liste des informations à écrire. Chaque langage a sa syntaxe propre, pour les algorithmes on les fait apparaître entre guillemets. L'algorithme ci-dessus s'écrira comme suit :

Variables A, B, C : **entier**

Debut

écrire « Nombre saisis », X , Y

Lire A, B

C ← A * B

écrire « Produit » ,C

Fin

L'algorithme va afficher le message suivant :

Nombre saisis

L'utilisateur va saisir deux nombres, par exemple -2 et 8

WWW.DZPEN.COM | قلم الجزائر

L'algorithme va effectuer le produit, puis il va afficher le résultat suivant :

Produit -16

Pour les deux autres versions, il suffit de remplacer les deux instructions d'écriture de commentaire par :

Ecrire X, « fois », Y , « égal », C

Ou bien

Ecrire X, « x », Y , « = », C

3.5 Résumé

L'algorithme se compose d'une suite d'ordres à exécuter qui sont couramment appelés instructions (ce sont les actions de l'analyse)

Nous avons vu que chaque variable est représentée dans l'algorithme par son identificateur. Nous avons vu que nous pouvant associer à un identificateur un nombre, valeur à l'étape considérée de la variable qu'il représente. Nous avons appelé cette action **l'affectation.**

Mais l'affectation ne nous permet pas d'initialiser les variables, c'est pourquoi vous pouvez utiliser l'instruction de lecture exprimée par : **lire** nom de la variable.

Pour éditer les résultats à l'écran, il est indispensable d'utiliser une instruction d'écriture exprimée par : **écrire** nom de variable.

Cependant, pour l'éditer les résultats à l'écran, il est préférable d'expliciter le résultat numérique par un **message** (**commentaire**). Tous les messages encadrés par des guillemets apparaissent à l'écran tel qu'ils ont été tapés.

3.6 Exercice de synthèse

Un enfant à l'école fait l'apprentissage de la multiplication, aidez-le à trouver le carré d'un nombre en écrivant l'algorithme adéquat. (ajouter des commentaires pour le rendre plus clair).

5 ⟶ 5*5 = 25

WWW.DZPEN.COM | قلم الجزائر

Chapitre 4 :

Ecriture d'un algorithme qui utilise les structures de contrôle

Les algorithmes décrits dans le chapitre précédant étaient très simples et une calculatrice aurait suffi pour leur exécution. En effet, pour l'instant, nous sommes seulement en mesure de décrire une suite d'opérations, chacune devant être exécutée une et une seule fois, dans l'ordre où elles étaient écrites (exécution séquentielle). Or, la puissance d'un algorithme provient du fait qu'il peut effectuer des choix dans la façon d'exécuter les instructions.

Il a été démontré que **pour représenter n'importe quel algorithme**, il faut disposer des possibilités suivantes :

- La **structure de séquence** qui indique que les opérations doivent être exécutées les unes après les autres.
- La **structure de choix** qui indique quel ensemble d'instructions doit être exécutée suivant les circonstances.
- La **structure de répétition** qui indique qu'un ensemble d'instructions doit être exécuté plusieurs fois.

Jusqu'à présent, nous avons décrit la structure de séquence. Nous décrirons dans ce chapitre les structures de choix : l'alternative et le choix multiple, et les structure de répétitions.

4.1 Définition des structures de contrôles

4.1.1 Définition

Une structure de contrôle sert à contrôler le déroulement d'un traitement.

Un traitement peut s'exécuter de différentes manières :

- Séquentiellement (vu dans le chapitre précédant).
- Alternativement.
- Répétitivement.

4.2 Le choix

4.2.1 Alternative

4.2.1.1 La structure alternative « Si …… alors »

La structure alternative simple « Si …… alors » s'écrit sous la forme suivante :

WWW.DZPEN.COM | قلم الجزائر



Si la condition est vraie alors la séquence d'instruction s'exécute Sinon ne rien faire.

Exemple 1 :

Si demain il pleuvrait alors j'annulerai ma visite.

Condition est : « demain il pleuvrait »

La séquence est : « annulerai ma visite »

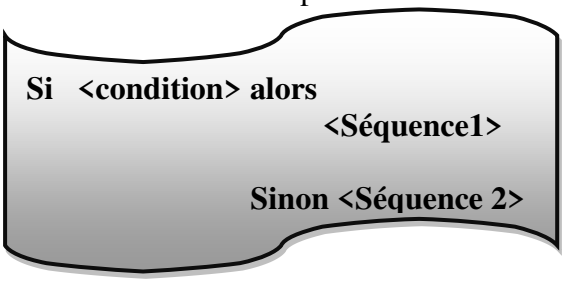
Exemple 2 :

Si A > B **alors** **Séquence**

Condition **écrire** « A est supérieur à B »

4.2.1.2 La structure alternative « Si …….. alors …….. Sinon »

La structure alternative composée « Si …… alors » s'écrit sous la forme suivante :



Si la condition est vérifiée alors séquence 1 s'exécute et si elle n'est pas vérifiée, c'est la séquence 2 qui va s'exécuter.

Exemple 1 :

Si demain il pleuvrait **alors** j'annulerai ma visite.

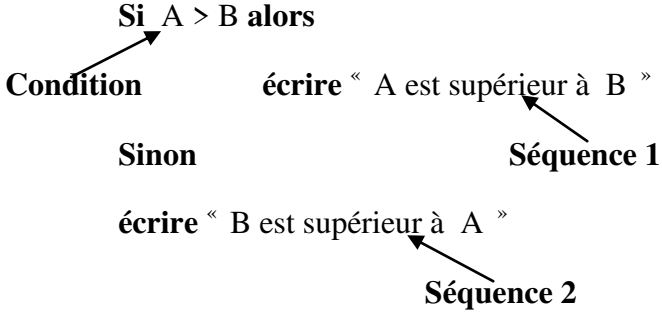
Condition **Séquence1 (condition Vrai)**

Sinon je visiterai mes parents en compagne.

Séquence2 (condition fausse)

WWW.DZPEN.COM | قلم الجزائر

Exemple 2 :



Exemple :

Dans une banque chaque compte client à un solde, considérons un algorithme qui doit produire un message selon la valeur de la variable nommée **solde** comme suit :

- Si la valeur de solde est positive, l'algorithme doit afficher :

  « Solde positif »

- Si la valeur de solde est négative ou nulle, l'algorithme doit afficher :

  « Solde négatif ou nul»

Nous écrirons ce choix comme suit :

```
Si  solde > 0 alors
                écrire « Solde positif »
Sinon
        ecrire « Solde négatif ou nul »
```

Si la condition (**solde > 0**) qui suit le mot « «**Si** » est vraie, l'algorithme exécute l'instruction « **écrire** « solde positif » » qui figure après le « **alors ».**

Si la condition est fausse, il exécute l'instruction qui figure après le « **sinon** », soit « **écrire** « Solde négatif ou nul » ».

L'écriture ci-dessus est équivalente aux deux écritures suivantes :

WWW.DZPEN.COM | قلم الجزائر

Si solde > 0 **alors écrire** « Solde positif »

Sinon

écrire « Solde négatif ou nul »

Ou bien :

Si solde > 0 **alors**

écrire « Solde positif »

Sinon écrire « Solde négatif ou nul »

Dans l'exemple précédent, il n'y a qu'une seule instruction dans chaque choix. Il peut bien sûr y avoir plusieurs instructions dans chaque choix, comme dans l'exemple ci-dessous :

Variables nombre, C, T, choix : **entier**

écrire « donnez un nombre : »

lire nombre

écrire « Effectuez un choix : 1. Carré 2.Triple »

lire choix

si choix = 1 alors

C ← nombre * nombre

écrire « Le carré de », nombre, « est : » ,C

sinon

T ←3 * nombre

écrire « Le triple de », nombre, « est : » ,T

écrire « fin de l'algorithme »

Par convention ici également, nous écrirons les instructions faisant partie de chaque choix décalées vers l'intérieur. L'instruction « **écrire** » « fin du l'algorithme » » ne fait donc pas partie du choix **sinon.**

WWW.DZPEN.COM | قلم الجزائر

Exemple d'exécution :

```
Donner  un nombre
7
Effectuez un choix : 1.Carré  2.Triple
2
Le triple de 7 est : 21
fin du programme
```

Dans cet exemple, la condition « choix = 1 » est fausse, l'algorithme exécute donc le **« sinon »**. Il passe ensuite à l'instruction suivante la structure de choix, c'est-à-dire : « **écrire** " fin du l'algorithme " **».** Cette instruction est exécutée dans tous les cas.

Considérons l'exemple d'exécution suivant :

```
 Donner  un nombre
9
Effectuez un choix : 1.Carré  2.Triple
1
Le triple de 9 est : 81
fin du programme
```

Dans le cas, la condition « choix = 1 » est vraie. L'algorithme l'exécute donc, puis passe ensuite à l'instruction suivant la structure de choix, c'est-à-dire :
« **écrire** " fin du l'algorithme " **».**

Les conditions dans la structure choix

Dans la structure, les conditions peuvent être simples ou complexes.

Les conditions simples

Une condition simple consiste en la comparaison de deux expressions de même type.

Exemples : si x>5 alors

Si c= " A " alors

WWW.DZPEN.COM | قلم الجزائر

Dans les deux conditions, nous comparons deux variables de même type : numérique dans le 1er cas, et caractère dans le 2ème.

Les symboles de comparaison usuels sont les suivants :

Symbole	signification
=	Egal à
<	Inférieur à
>	Supérieur à
<=	Inférieur ou égal à
>=	Supérieur ou égal à
< >	Différent de

Exemple :

L'algorithme ci-dessous permet de lire deux lettres et de dire si elles sont ou non dans l'ordre alphabétique :

Variables lettre1, lettre2 : **caractère**

écrire « Entrez deux lettres de l'alphabet : »

lire lettre 1, lettre 2

si lettre 1<= lettre 2 **alors**

écrire « Elles sont dans l'ordre alphabétique »

Remarque :

Si au lieu de comparer deux lettres on compare deux nombres pour savoir s'ils sont ou non dans l'ordre croissant, l'algorithme s'écrira de la même façon que ci-dessus :

Variables nb1, nb2 : entier

écrire « Entrez deux nombres : »

lire nb1, nb2

si nb1 <= nb2 **alors** **écrire** « ils sont dans l'ordre croissant »

WWW.DZPEN.COM | قلم الجزائر

sinon écrire « ils ne sont pas dans l'ordre croissant»

Les conditions complexes

Une condition est dite complexe lorsqu'elle est formée de plusieurs conditions simples reliées entre elles par des opérations logiques **« et »** et

« ou ». les symboles représentant ces opérations dépendront du langage utilisé.

Exemples :

Si X > 0 **et** X < 10 **alors**

Si X > 0 **ou** Y > 0 **alors**

Dans le 1er cas, la condition est vraie si les deux conditions simples X > 0 et X < 10 sont vraies. Il suffit que l'une des deux soit fausse pour la condition soit fausse.

Dans le 2ème cas, la condition est vraie si l'une au moins des deux conditions simples X > 0 ou Y > 0 est vraie.

Remarque :

Il existe un opérateur de négation « **non** ». Par exemple, la condition non (X > 0) est vraie si la condition X > 0 est fausse, et inversement. La condition non (X >0) est équivalente à (X <= 0)

4.2.2 Choix multiple

4.2.2.1 La structure alternative « Cas où »

```
Cas < variable > où
      Cas 1 : Séquence 1
      Cas 2 : Séquence 2
      ------  : ------------
      Cas n : Séquence n
            Sinon
                  Séquence
```

Principe :

Cette structure de contrôle présente plusieurs cas d'exécution du traitement, mais un seul cas sera exécuté.

WWW.DZPEN.COM | قلم الجزائر

Remarque :

La structure **cas …où** peut ne avoir de **sinon.**

Exemple 1 :

Selon le cas qui se présente je prendrai la décision qu'il faut :

Cas 1 : Mon chiffre d'affaires est inférieur à 600000 DA alors aucun investissement n'est à envisager.

Cas 2 : Mon chiffre d'affaires atteint les 1000000 DA alors j'achèterai un véhicule utilitaire.

Cas 3 : Mon chiffre d'affaires dépasse les 1500000 DA alors j'achèterai un véhicule utilitaire et autre de service.

Exemple 2 : Afficher les noms des jours de la semaine :

Variable j :entier

Lire j

Cas j **où**

1 : écrire « C' est Samedi »

2 : écrire « C' est Dimanche »

3 : écrire « C' est Lundi »

4 : écrire « C' est Mardi »

5 : écrire « C' est Mercredi »

6 : écrire « C' est Jeudi »

7 : écrire « C' est Vendredi »

Fin

4.2.3 Exercices

Exercice 1 :

Dans un magasin, le stock est gérer par un magasinier qui calcule le montant de la facture par produit acheté.

Il calcule le montant de la facture à partir du prix unitaire et de la quantité.

WWW.DZPEN.COM | قلم الجزائر

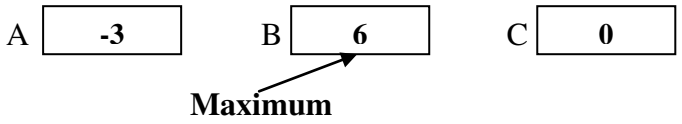
Si le montant dépasse 500DA, le magasinier fait une remise de 3%.

Ecrit l'algorithme qui va aider ce magasinier à calculer le montant de la facture.

Exercice 2 :

On demande à un élève de déterminer le maximum de trois nombres entiers.

Trouvez l'algorithme qui permet de comparer les trois nombres.



Exercice 3 :

On vous apprend dès l'apprentissage de la longue française que les lettres se divisent en voyelles et en consonnes.

Les voyelles sont :

'a' , 'e' , 'i' , 'u' , 'o' , 'y'

Ecrire l'algorithme qui lit un caractère et affiche si c'est une voyelle ou une consonne.

- On utilise la structure alternative **Si ….. alors.**

- On utilise le choix multiple **cas …. où.**

4.3 La répétition

Nous avons vu qu'un algorithme peut effectuer des choix dans la façon d'exécuter les opérations.

Dans cette partie, nous allons voir qu'un algorithme peut répéter plusieurs fois les mêmes opérations.

Par exemple, un algorithme de calcul de la paye répétera pour chaque employé les mêmes opérations pour établir une fiche de paye.

S'il est théoriquement possible de se contenter d'une seule structure de répétition, bien choisie, pour exprimer tous les algorithmes, l'expérience a montré l'utilité d'en

WWW.DZPEN.COM | قلم الجزائر

définir plusieurs, chacune à des circonstances particulières. Ce sont les boucles **tant que**, **répéter . . . jusqu'à** et **pour**.

4.3.1 Répétition contrôlée par des conditions.

Les boucles concernées par la répétition contrôlée par des conditions sont :

La boucle **tant que** et la boucle **répéter . . . jusqu'à.**

Ces boucles sont à utiliser dès qu'il faut faire un même traitement plusieurs fois et qu'on ne sache pas, au départ, combien de fois on répétera le traitement. Pour cela on utilisera les conditions.

4.3.1.1 La boucle « Tant que »

Exemple introductif :

Nous avons le nom d'une personne (nom_donné) et on veut connaître son adresse et nous avons une liste de noms de personnes qu'on compare un à un avec le nom_donnée et en cas d'égalité, afficher l'adresse de cette personne.

Debut

Lire nom_donné

Lire nom1

Si nom1 = nom_donné **alors écrire** adresse1

Sinon lire nom2

Si nom2 = nom_donné **alors écrire** adresse 2

Sinon lire nom3

Si nom3 = nom_donné **alors** . . . .

L'inconvénient de cet algorithme tient au fait que l'auteur ne sait pas quand il doit s'arrêter d'écrire. Plus précisément, il lui est impossible de savoir combien de fois il doit écrire l'instruction de comparaison au nom_donné.

Cet exemple montre que la séquence et l'alternative ne sont pas en elles-mêmes suffisantes pour exprimer des algorithmes dont la longueur peut varier selon les circonstances. Il est donc nécessaire d'introduire le moyen de répéter certaines opérations d'un algorithme un nombre quelconque de fois. La structure qui permet cela est appelée structure **répétitive.**

WWW.DZPEN.COM | قلم الجزائر

La formulation générale de la boucle **Tant que** est :

Tant que < condition vérifiée > **faire**

< Séquence >

Fin tant que

Le principe est que tant que la condition est vérifiée la séquence s'exécute, elle ne s'arrêtera que lorsque la condition est invérifiable.

Exemple 1:

Tant que le clou n'est pas totalement enfoncé, frapper avec le marteau.

- **Opérations répétées** : frapper avec le marteau.
- **Condition d'arrêt** : clou totalement enfoncé.

Exemple 2:

Afficher les 100 premiers nombres entiers positifs.

Variable I : entier

I ← 0 ← **compteur initialisé**

Tant que I < = 100 **faire**

I ← I + 1 ← **condition d'arrêt**

écrire I ← **incrémentation du compteur**

Fin

Eléments de la boucle tant que :

- **La condition d'arrêt :**

Sans condition d'arrêt la boucle devient infinie.

Dans l'exemple 2 (I > 100) est la condition d'arrêt.

- **Le compteur de la boucle :**

Il sert à faire avancer la boucle (incrémentation).

WWW.DZPEN.COM | قلم الجزائر

Dans l'exemple 2 (I⟵I + 1) est l'instruction qui permet d'incrémenter la boucle.

- **Initiation du compteur :**

La condition d'arrêt porte sur le compteur de la boucle 'I', pour cela, la boucle doit commencer par une valeur initiale (I = 0)

4.3.1.2 La boucle « répéter … jusqu'à »

Comme la boucle « Tant que », ce type de répétition est utilisé lorsque le nombre de fois que la séquence d'instructions à répéter est inconnu au moment où cette séquence est abordée pour la première fois mais le corps de la boucle est **toujours exécuté au moins une fois**.

La formulation générale de la boucle **répéter . . . jusqu'à** est :

Répéter

< Séquence >

Jusqu'à < Condition vérifiée >

Exemple 1 :

Frapper avec le marteau, **Répéter** ….. **jusqu'à** avoir le clou totalement enfoncé,.

- **Opérations répétées** : frapper avec le marteau.
- **Condition d'arrêt** : clou totalement enfoncé.

Exemple 2:

Afficher les 100 premiers nombres entiers positifs.

Variable I : entier

I ⟵ 1 ⟵ **initialisation du compteur**

Répéter

écrire I

I ⟵ I + 1 ⟵ **incrémentation du compteur**

Jusqu'à I > 100 ⟵ **condition d'arrêt**

WWW.DZPEN.COM | قلم الجزائر

Fin

Eléments de la boucle Répéter:

- **La condition d'arrêt :**

  Celle-ci doit être définie d'une façon exacte pour arrêter l'exécution de la séquence (ici I > 100), la condition doit toujours porter sur le compteur de la boucle (dans le cas où le nombre d'itérations est connu).

- **Incrémentation :**

  Dans ce cas il y'a nécessité d'incrémenter le compteur de la boucle, car le nombre d'itérations est connu.

- **Initiation :**

Le compteur doit être initialisé à une valeur initiale pour débuter l'exécution.

Exemple 3:

Soit l'algorithme suivant :

Variables n, p : entier

Répéter

écrire « Donner un nombre : »

lire n

p ← n * n

écrire « Le carré de ce nombre est : », p

jusqu'à n = 0 ← **condition d'arrêt**

écrire « fin de l'algorithme »

Les instructions encadrées par les mots **répéter** et **jusqu'à** sont celles qu'il faut répéter jusqu' à ce que la condition n = 0 soit vraie.

Le nombre de répétitions de ces instructions n'est pas indiqué explicitement. Il dépend des données qui seront fournies à l'algorithme.

WWW.DZPEN.COM | قلم الجزائر

Exemple d'exécution :

Donner un nombre : 5

Le carré de ce nombre est : 25

Donner un nombre : 7

Le carré de ce nombre est : 49

Donner un nombre : 0

Le carré de ce nombre est : 0

fin de l'algorithme

4.3.2 Répétitions avec indice

Ce type de répétition est utilisé lorsque le nombre de fois que la séquence d'instruction à répéter est connu.

La boucle concernée par la répétition avec indice est la boucle **« Pour ».**

On utilise un compteur associé à la boucle, la valeur augmente de 1 à chaque passage dans la boucle. Donc il sert a compter le nombre de fois que la boucle a été exécutée.

4.3.2.1 La boucle « Pour »

La formulation générale de la boucle **Pour** est :

Pour < compteur > allant de < valeur initiale > à < valeur finale > faire

< Séquence >

- Pour compteur allant de valeur initiale jusqu'à finale exécuter la séquence d'instructions.
- Le nombre de fois que la séquence s'exécute est (valeur finale – valeur initiale + 1)

Exemple 1 : Mettre des étiquettes pour 100 produits.

Pour les 100 produits en chaîne, mettre une étiquette pour chacun.

- **Opérations répétées :** Mettre une étiquette.
- **Condition d'arrêt :** Finir les 100 produits.

WWW.DZPEN.COM | قلم الجزائر

Exemple2 : Afficher les 100 premiers nombres entiers positifs

Variable I : entier

Pour I **allant de** 1 **à** 100 **faire**

écrire I

Fin.

Eléments de la boucle **Pour** :

- **Valeur initiale / finale :**

  Le compteur de la boucle doit commencer le comptage à partir de la valeur initiale et finir avec la valeur finale. (ici Vi = 1, Vf = 100)

Remarque :

On voit clairement ici que la boucle Pour ne nécessite pas d'incrémentation ni d'initialisation au préalable.

Exemple 3 : calcule de la moyenne d'une liste contenant un nombre définit d'entiers, on appellera ce nombre N.

Variables R, S, I, N : Entier

M : Reel

écrire « Donner le nombre des entier : »

Lire N ⟵ Nombre d'entiers

S ⟵ 0 ⟵ Initialisation de la somme des entiers

Pour I **allant de** 1 **à** N **faire**

Saisie de l'entier ⟶ **Lire** R

Calcul de la somme des entiers ⟶ S ⟵ S + R

fpour

Moyenne ⟶ M ⟵ S / N

écrire « La moyenne est : », M

WWW.DZPEN.COM | قلم الجزائر

écrire « fin de l'algorithme »

4.3.3 Exercices

Exercice 1 :

Ecrire un algorithme qui permet d'afficher les lettres de l'alphabet, séparés par des étoiles (*)

Le résultat | **A*B*C*D*. . . . . .*X*Y*Z**

Exercice 2 :

Le factoriel d'un nombre N est égale à :

N ! = 1*2*3* . . . . . .(N-1)*N

N ! = N*(N-1)*. *2*1

Ecrire un algorithme qui calcule le factoriel d'un nombre entier.

4.4 Résumé

Au cours de ce chapitre, vous avez appris à utiliser les structures alternatives et les structures de répétitions.

Voici un résumé des principaux éléments vus au cours de ce chapitre.

La **première section** vous a donné la définition des structures de contrôles, et vous avez appris qu'un traitement peut s'exécuter de différentes manières :

- Séquentiellement.
- Alternativement.
- Répétitivement

Dans la **deuxième section** vous avez découvert que la puissance d'un algorithme provient du fait qu'il peut effectuer des choix dans la façon d'exécuter les instructions. La forme la plus simple d'une situation de choix est celle où deux décisions sont possibles dépondant d'une condition qui doit être une expression de type booléen.

Vous avez vu aussi le choix multiple qui est une généralisation de la structure alternative. Au lieu d'avoir deux possibilités, on a au moins trois, et le choix multiple

WWW.DZPEN.COM | قلم الجزائر

associe à chaque valeur une ou plusieurs actions.

La **troisième section** vous a fait découvrir la structure répétitive, qui permet à un algorithme de répéter plusieurs fois les mêmes opérations. Cela implique l'utilisation de la structure de répétition qui est **la boucle**, on appelle les boucles « **tant que »**, « **répéter . . . jusqu'à »,** la répétition contrôlée par des conditions, et on appelle la boucle « **Pour »** répétitions avec indice.

Pour la boucle « **tant que »** l'exécution de la répétition se fera si la condition d'accès à la boucle est vérifiée.

Pour la boucle « **répéter . . . jusqu'à »** l'exécution de la répétition se fera au moins une fois, ensuite s'arrêtera quand la condition se vérifiée.

Pour la boucle « **Pour »** elle s'exécute toujours, le nombre d'exécutions de la séquence est connu d'avance.

Toutes ces notions sont essentielles à l'élaboration d'algorithme complet qui peut prétendre à la résolution de n'importe quel problème.

4.5 Exercice de synthèse

Ecrire un algorithme qui permet d'entrer un entier, vérifier s'il est négatifs et l'afficher.

Remarque : répéter la lecture jusqu'à l'entrée d'un nombre positif (N > 0).

WWW.DZPEN.COM | قلم الجزائر

Chapitre 5 :

Ecriture d'un algorithme qui utilise des chaînes de caractères

On a vu que pour manipuler un caractère, il fallait utiliser le type **caractère,** or il arrive que l'on est besoin de manipuler non plus un simple caractère mais toute uns suite de caractères que l'on nomme les **chaînes de caractères.**

Les opérations sur les chaînes de caractères sont :

- Déclaration des variables de chaîne de caractère de longueur variable.
- La lecture et l'écriture de chaîne.
- L'affectation entre chaîne.
- Existence d'opérateurs de comparaison et de concaténation.
- Un jeu complet de fonctions et de procédures portant sur les chaînes de caractères.

5.1 Déclaration

5.1.1 Définition et déclaration

Définition : une chaîne de caractères est une suite de caractères alphabétiques (lettres majuscules ou minuscules), de caractères numériques (lettres de 0 à 9) ou d'autres symboles tels que les opérateurs (+, - , *, /) ou les signes de ponctuation …

Exemples de chaîne de caractères :

'1'

'B'

'ab'

'772_42_12'

'251232'

L'espace entre deux caractères, parfois indiqué par (_) pour plus de clarté, est lui aussi considéré comme un caractère.

Comme le montrent les exemples précédents une chaîne de caractères peut avoir une longueur quelconque

1. Une seule lettre est considérée comme une chaîne de longueur 1.
2. Une phrase est une chaîne ayant pour longueur le nombre de caractères contenus dans la phrase, espaces compris.
3. Les chaînes de caractères peuvent être des constances, nous avons déjà rencontré des exemples sous forme de message, délimité par des guillemets.

WWW.DZPEN.COM | قلم الجزائر

Par exemple « tapez un nombre positif ».

Déclaration d'une chaîne de caractères

La déclaration d'une chaîne de caractères s'exprime par :

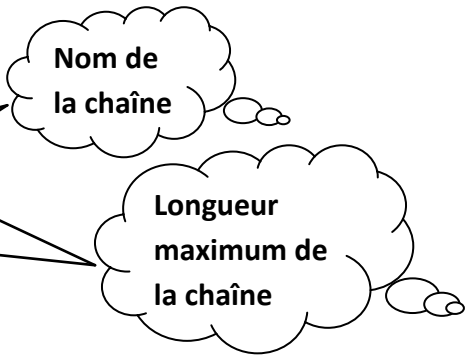
Chaine identificateur de la chaine (longueur maximum de la

Exemple :

Variables

Chaine titre (6)

Chaine SS (13)



5.1.2 Exercice

1. Faire la déclaration d'une chaîne de caractères nommé phrase de longueur maximum 20
2. Citez les différentes opérations qu'on peut effectuer sur une chaîne de caractères.

5.2 Opérations sur les chaînes de caractères.

5.2.1 La lecture et l'écriture

La lecture :

Les chaînes de caractères, étant des variables, obéissent aux mêmes règles que les variables numériques pour la lecture.

Pour lire une variable de type chaîne on utilise la structure suivante :

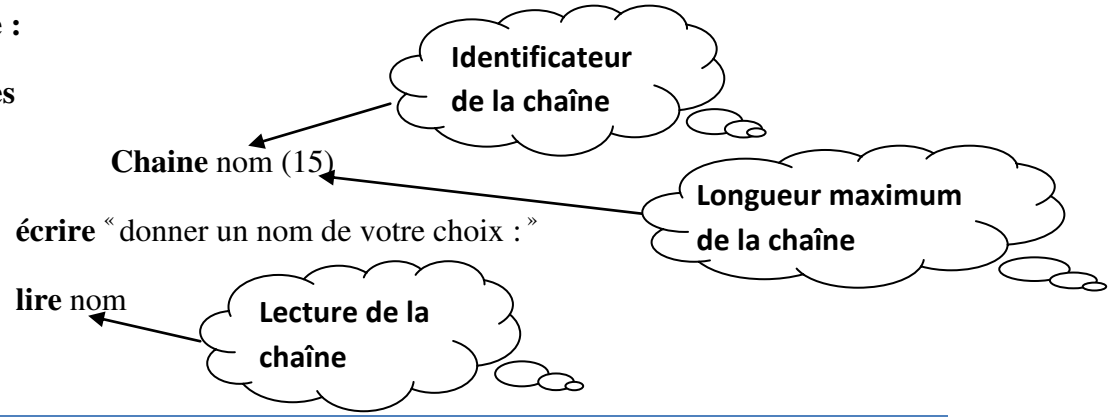
Exemple :

Variables

Chaine nom (15)

écrire « donner un nom de votre choix : »

lire nom



WWW.DZPEN.COM | قلم الجزائر

L'écriture :

L'affichage de chaîne de caractères ne pose aucun problème particulier, il est possible d'imposer un gabarit qui sera respecté, s'il est au moins égale à la longueur courante de la chaîne.

Pour écrire une variable de type chaîne on utilise la structure suivante :

Variables

Chaine mot (10)

. . . . . .

écrire « Le résultat est : » , mot

5.2.2 L'affectation

L'affectation se fait entre deux variables de type chaine.

Exemple :

Variables **Chaine** Mot1(20)

Chaine Mot2 (5)

Mot1 ⟵ Mot2 **cette opération est juste**

Mot2 ⟵ Mot1 **cette opération est fausse**

On peut aussi affecter une chaîne

Exemple :

Mot ⟵ 'Bonjour' Mot **Bonjour**

Mot2 ⟵'Monsieur' Mot2 **Monsieur**

Mot1 ⟵ '' **affectation d'une chaîne vide.** Mot1

5.2.3 La comparaison

Les comparaisons de chaînes sont simplement basées comme une comparaison de caractères par l'ordre du code ASCII de chaque caractère qu'il est constitué.

L'égalité des chaînes a lieu lorsqu'ils ont la même longueur courante et qu'elles sont constituées des mêmes suites des caractères.

WWW.DZPEN.COM | قلم الجزائر

Exemple :

'ALGO' < 'ALGO1'

'Pages' = 'Pages'

'2' > '12'

5.2.4 La concaténation

Cette opération consiste à juxtaposer deux ou plusieurs chaînes pour former une seule chaîne, elle peut être obtenue soit avec l'opérateur ' + ' ou bien avec la fonction **CONCAT**.

Concat (chaîne_1, chaîne_2, ..., chaîne_n)
chaîne_1 + chaîne_2 + ... + chaîne_n

Exemple :

Mot1 ⟵ 'Bonjour'

Mot2 ⟵ ' '

Mots3 ⟵ 'monsieur'

Mot ⟵ Mot1 + Mot2

Ou

Mot ⟵ Concat(Mot1, Mot2)

Le résultat est : Bonjour monsieur

5.2.5 Les fonctions et les procédures portant sur les chaînes

5.2.5.1 La fonction « Length »

Elle permet d'obtenir la longueur **courante** de la chaîne c'est-à-dire le nombre de caractères la constituant y compris les blancs et les caractères non affichables.

WWW.DZPEN.COM | قلم الجزائر

Exemple :

Mot1 ← 'Bonjour'

Length (mot1) ← **Le résultat de la fonction est : 7**

Mot ← 'sidi bel abbès'

Length (mot) ← **Le résultat de la fonction est : 14**

5.2.5.2 La fonction d'extraction de sous–chaînes : « COPY »

Elle permet d'extraire une partie de la chaîne (une sous-chaîne) de longueur donnée à partir d'une position donnée.

COPY (chaîne, position, longueur de la sous-chaîne)

Exemple :

Mot1 ← 'Bonjour'

Mot ← 'classe'

CH ← COPY (Mot , 2 , 2)

écrire (CH) ← **Le résultat de la fonction est : 'la'**

écrire (COPY(Mot1 , 2 , 4)) ← **Le résultat de la fonction est : 'onjo'**

écrire (COPY(Mot1 , 4 , 4)) ← **Le résultat de la fonction est : 'jour'**

Exemple : Utilisation de la fonction COPY

Variables

chaine Mot (15)

I : entier

écrire « Donnez un mot »

WWW.DZPEN.COM | قلم الجزائر

lire Mot

pour I allant de 1 jusqu'à Length (mot) faire

écrire (COPY(Mot , 1 , I))

Exécution de l'algorithme :

Mot	I	**Affichage**
TURBO		
	1	T
	2	TU
	3	TUR
	4	TURB
	5	TURBO

Remarque :

- Si on veut extraire plus de caractères qu'il n'est possible, on obtiendra que la partie existante effectivement.

Exemple :

Mot ⟵ 'Bonjour'

CH ⟵ COPY (Mot , 6 , 4) ⟵ **Le résultat de la fonction est : 'ur'**

- Si la position indiquée est située or de la chaîne, on obtiendra une chaîne vide (la longueur est vide)

Exemple :

Mot ⟵ 'Bonjour'

CH ⟵ COPY (Mot , 9 , 2) ⟵ **Le résultat de la fonction est : '' (la chaîne vide)**

WWW.DZPEN.COM | قلم الجزائر

- Si la position est supérieure à 255, on aura une erreur d'exécution.

5.2.5.3 La fonction de localisation de sous–chaînes « POS »

Cette fonction permet de situer une sous-chaîne dans une chaîne.

Dans le cas où elle s'y trouve la fonction **POS** fournie le rond (la position) du caractère où commence la sous-chaîne.

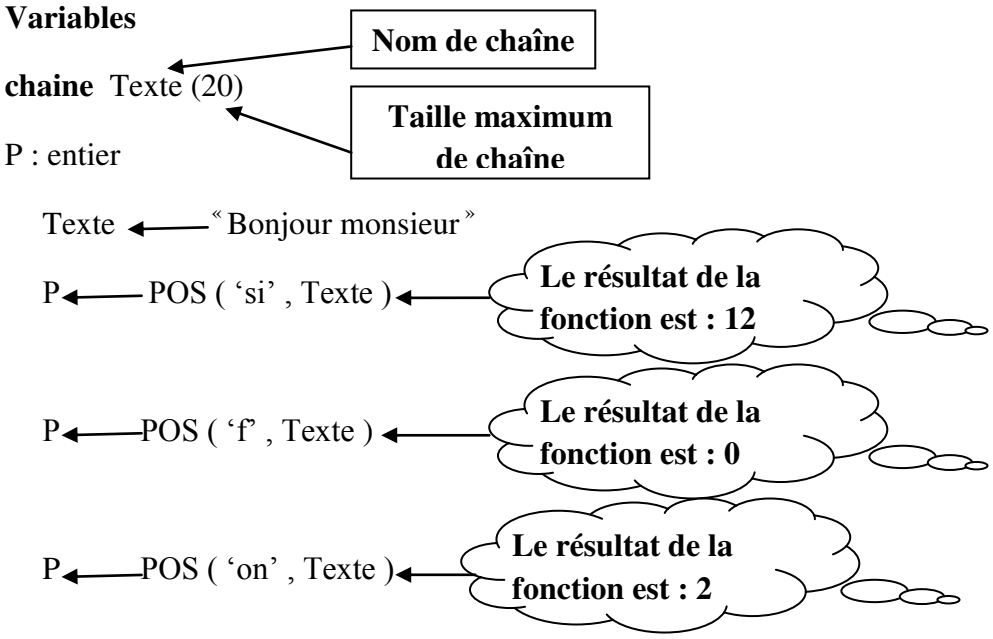
Dans le cas contraire elle fournit 0.

POS (sous-chaîne , chaîne)

Remarques :

- La fonction **POS** renvoi 0 si la sous-chaîne n'existe pas.
- La fonction **POS** renvoi un nombre différent de 0 si la sous-chaîne existe.

Exemple :



WWW.DZPEN.COM | قلم الجزائر

5.2.5.4 La procédure de suppression de sous–chaîne dans une chaîne « DELETE »

Elle permet de supprimer un ou plusieurs caractères à partir d'une position donnée.

Donc la chaîne est modifiée.

DELETE (chaîne, position, longueur de la sous-chaîne)

Variables

chaine Texte (20)

Texte ← « Bonjour monsieur »

DELETE (texte , 8, 9)

écrire (texte) ← **Le résultat de la fonction est:'Bonjour'**

5.2.5.5 La procédure de conversion d'une chaîne en numérique « VAL »

Permet de convertir lorsque cela est possible une chaîne de caractères en une valeur numérique. Entière ou réelle selon le tupe de variable mentionner pour le résultat.

Si la conversion n'est pas possible **VAL** renvoi dans un indicateur la position du 1er caractère de la chaîne qui a empêché la conversion d'aboutir.

L'indicateur est 0 si la conversion s'est dérouler normalement.

VAL (chaîne , le résultat , l'indicateur)

- l'indicateur → **0** : la conversion est exécutée
  Différent de 0 : la conversion non exécutée.
  Position du caractère ayant causé un problème.
- le résultat → Valeur numérique dans le cas où l'indicateur = 0
- chaîne → Chaîne a convertir.

WWW.DZPEN.COM | قلم الجزائر

Exemple : l'algorithme qui permet une lecture d'un nombre entier en toute sécurité, c.-à-d contrôler la lecture des entiers au clavier pour éviter les erreurs de saisie.

Variables

chaine CH (20)

N , IND : entier

écrire « Donner un nombre entier »

Répéter

Lire CH

VAL (CH , N , IND)

Si IND < > 0 **alors** écrire « entier s'il vous plait »

Jusqu'à IND = 0

écrire « Merci pour : », N

Lecture de la chaîne constitué

La conversion est non exécutée

La conversion exécutée

Valeur numérique

5.2.5.6 La procédure de conversion du numérique en chaîne « STR »

Elle permet de convertir un entier ou réel en une chaîne de caractères. Il est possible d'agir sur le format du résultat ainsi obtenu à l'aide du gabarit de sortie.

STR (Valeur numérique , chaîne résultat)

Variables

chaine CH (20)

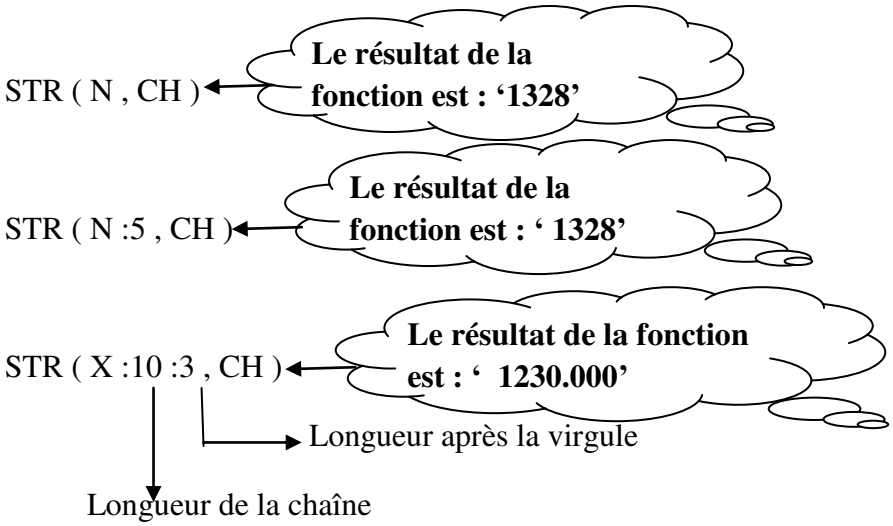
N: entier

X : réel

N⟵1328

X⟵1.23 E3

WWW.DZPEN.COM | قلم الجزائر



5.2.5.7 La fonction insérer des caractères dans une chaîne

« INSERT »

Elle permet d'insérer une sous-chaîne dans une chaîne a une position de début données.

INSERT (sous-chaîne , chaîne ,position-début)

Exemple :

Insert (' Les ' , texte, 1)

Si texte à pour valeur : 'Belles maisons' ← **Le résultat sera : 'Les Belles maisons'**

5.2.6 Exercice

Exercice 1 :

Ecrire l'algorithme qui permet de compter le nombre d'apparition de chaque voyelle dans un texte données.

Exercice2 :

La lecture d'un nombre entier et qui l'affiche sur 7 caractères en remplaçant tous les blancs par des '*'.

Exemple :

125 donne : ' ****125'

-58451 donne :'*-58451'

WWW.DZPEN.COM | قلم الجزائر

5.3 Résumé

Ce chapitre vous a permis de voir que pour manipuler des caractères et du texte, on a introduit une nouvelle structure de donnée, qui est **les chaînes de caractères.** Voici, résumés ici les principaux éléments que vous avez appris dans ce chapitre.

A **la première section**, vous a exposé la définition de la chaîne de caractères, vous avez vu qu'elle est une suite de caractères quelconque d'une longueur maximum ne dépassant pas 255 caractères, et que vous pouvez et vous devez la déclarer avant de l'utiliser dans un algorithme, de la manière suivante :

Chaine identificateur de la chaine (longueur maximum de la chaîne).

Dans la **deuxième section** vous avez vu que les chaînes de caractères sont manipulées avec les mêmes instructions générales tels que la lecture, l'écriture, l'affectation, etc… que les variables numériques.

Vous pouvez aussi effectuer des opérations sur les chaînes de caractères comme :

- Les comparaisons de chaînes.
- la fonction CONCAT qui consiste à mettre bout à bout deux chaînes de caractères.
- Un ensemble de fonctions et de procédures portant sur les chaînes, comme la fonction « Length », la fonction d'extraction de sous–chaînes « COPY », la fonction de localisation de sous–chaînes « POS », etc.

5.4 Exercice de synthèse

Ecrire l'algorithme qui permet d'afficher successivement les chaînes obtenues par juxtaposition des éléments symétriques en accroissant a chaque fois leurs longueurs de 1, à partir d'un mot fourni en entrée.

Exemple :

Le mot est : TURBO

Le résultat : TO

TUBO

TURRBO

TURBURBO

TURBOTURBO

WWW.DZPEN.COM | قلم الجزائر

Chapitre 6 :

Ecriture d'un algorithme qui utilise des tableaux à une dimension

Dans les chapitres précédents nous avons manipulé les nombres individuellement, chaque nombre étant reconnu par un identificateur.

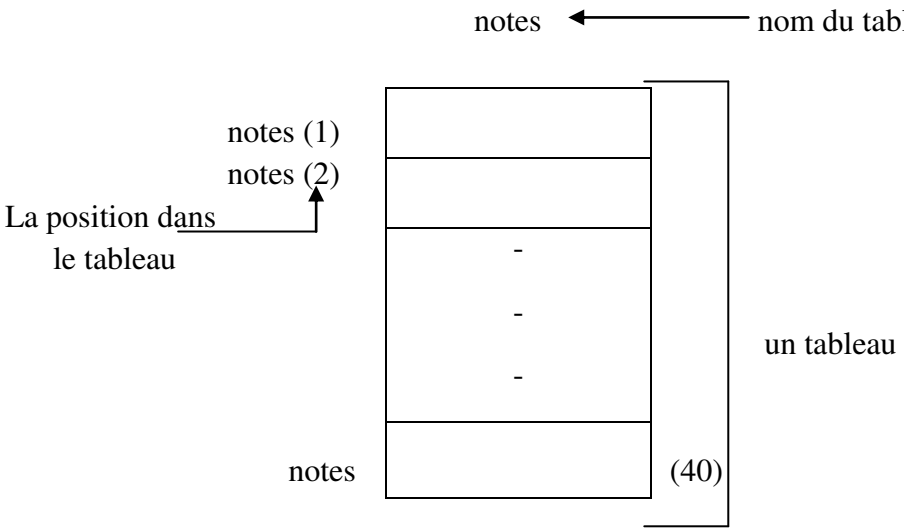
Dans le présent chapitre nous montrons l'intérêt qu'il peut avoir à regrouper sous le même identificateur plusieurs variables de même type (numérique, caractère, . . .) associées à un même nom. Nous obtenons ainsi un « tableau ».

Deux questions se posent tout de suite : Quelle est l'utilité d'avoir un ensemble de variables avec le même nom ? Et comment fera-t-on la différence entre des variables ayant le même nom ?

Supposons que vous ayez besoin, dans un programme, de manipuler et de conserver les notes de 40 élèves. Vous devez, par conséquent, déclarer 40 variables que vous nommez par exemple note1 , note2 , . . ., note40.

Il serait donc plus facile d'utiliser un seul nom, par exemple notes. Un seul nom vaut mieux que 40, cela va de soit. Mais comment les différencier ?

La réponse est dans la notion même de tableau: chaque élément du tableau est repéré par un numéro. C'est ce qui permet de différencier une variable d'une autre. En effet, elles ont toutes le même nom (qui est commun aux autres variables) suivi par un numéro entre parenthèses.



WWW.DZPEN.COM | قلم الجزائر

6.1 Déclaration d'un tableau : Vecteur.

6.1.1 Définition et déclaration

Définition :

Les tableaux à une dimension sont utilisés pour représenter une variable qui peut prendre plusieurs valeurs en même temps.

Un tableau regroupe des éléments de même type.

Un tableau T de longueur N est une suite de N variables, l'indice I étant un entier compris entre 1 et N. On accède à une valeur du tableau en indiquant son indice dans le tableau.

Exemple :

Prenons par exemple les moyennes suivantes : 12, 13, 14, 15, 16.5

On obtiendra le vecteur suivant :

V (1)	12	13	14	15	16.5
	V(1)	V(2)	V(3)	V(4)	V(5)

'I' est l'indice des cases du vecteur variant de 'I' jusqu'à '5' ; alors pour accéder à un élément de ce vecteur il suffit d'indiquer sa position. Par exemple, V(5) représente le 5ème élément de ce tableau qui est 16.5 .

Déclaration de tableau:

La déclaration des tableaux s'exprime par :

Tableau <nom du tableau> (dimension) : **type**

Exemple 1 :

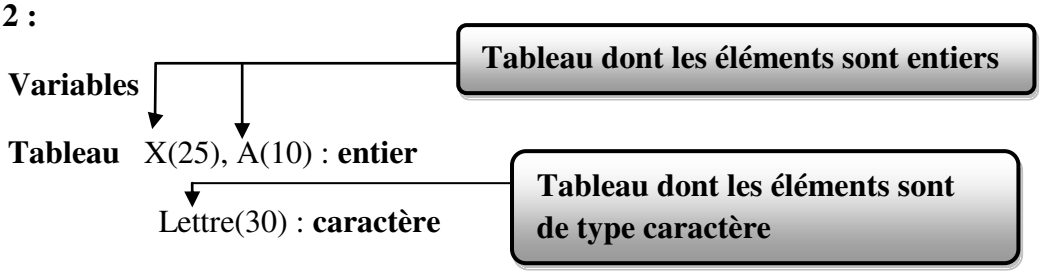
Variables

Tableau notes(40) : **entier**

Dans cet exemple, nous avons déclaré un tableau de type entier, nommé notes, et de dimension 40. Ce tableau pourra contenir 40 valeurs entières : notes (1), notes (2), . . . , notes (40).

WWW.DZPEN.COM | قلم الجزائر

Exemple 2 :



Signifie : déclaration de deux tableaux X et A de type entier, de dimensions respectives 25 et 10, et un tableau Lettre de type caractère de dimension 30.

6.1.2 Exercice

Exercice 1 :

Voici le tableau vect :

1	**2**		**19**	**20**
5	**6.2**	. . . . .	**-3**	**-7.55**

Faire la déclaration de ce tableau nommé VECT.

Exercice 2 :

Créer un tableau **TABMOIS** contenant le nombre de jour de chacun des moins d'une année non bissextile.

1	2	3	4	5	6	7	8	9	10	11	12
31	28	31	30	31	30	31	31	30	31	30	31

6.2 Manipulation d'un tableau : Lecture, écriture.

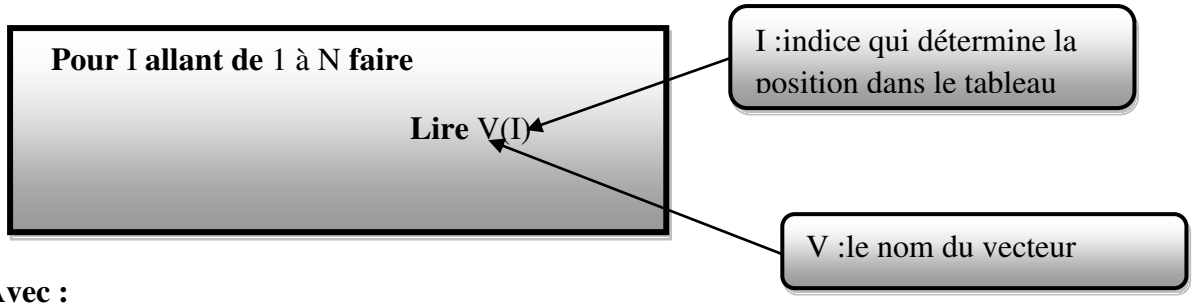
Une fois déclaré, un tableau peut être utilisé comme un ensemble de valeur simple. Les manipulations de base sont la lecture et l'écriture.

6.2.1 La lecture d'un tableau

Il est possible d'affecter des valeurs aux éléments d'un tableau lors de l'exécution d'un programme. Les valeurs sont alors saisies par l'utilisateur à la demande du programme.

Pour initialiser tous les éléments d'un tableau par des valeurs saisies par l'utilisateur, on écrira :

WWW.DZPEN.COM | قلم الجزائر



Avec :

V : la variable vecteur.

I : un indice qui indique la position d'une variable dans le vecteur V.

N : le nombre d'élément du vecteur (ou la dimension du tableau)

Exemple :

Pour remplir un tableau de 5 éléments et de type entier.

Variables I : **entier**

Tableau Prix (5) : **entier**

Pour I **allant de** 1 à 5 **faire**

écrire « Donnez un prix »

Lire Prix(I)

Exemple d'exécution :

Donnez un prix : 10
Donnez un prix : 8
Donnez un prix : 13
Donnez un prix : 45
Donnez un prix : 90

L'exécution de ce programme permet d'initialiser le tableau Prix comme suit :

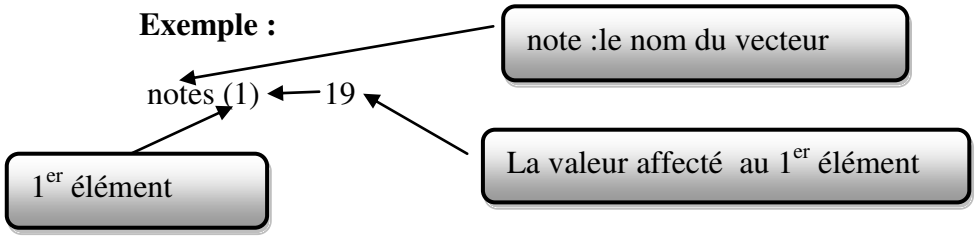
WWW.DZPEN.COM | قلم الجزائر

Prix

10	Prix(1)
8	Prix(2)
13	Prix(3)
45	Prix(4)
90	Prix(5)

On peut aussi assigner directement une valeur à un élément d'un tableau.

Exemple :



Cette écriture signifie :assigner au 1er élément du tableau notes la valeur 19.

- Pour initialiser les éléments d'un tableau on déclare par exemple deux tableaux X et Y de type entier, de dimensions respectives 10 et 15.

Variables I : **entier**

Tableau X (10), Y (15) : **entier**

Déclaration du tableau X et Y

Pour I **allant de** 1 à 10 **faire**

X(I) ← 1

1	1	1	1	1	1	1	1	1	1

Pour I **allant de** 1 à 15 **faire**

Y(I) ← 2

2	2	2	2	2	2	2	2	2	2

Dans cet exemple on initialise tous les éléments de X a 1 et tous les éléments de Y a 2.

- Si les deux vecteurs sont de même type, alors on peut affecter l'un à l'autre.

Tableau V1 (10) : **réel**

V2 (5) : **réel**

WWW.DZPEN.COM | قلم الجزائر

V1 ⟵ V2 cette opération est juste

V2 ⟵ V1 cette opération est fausse

6.2.2 L'écriture d'un tableau

De façon analogue à la lecture, l'écriture d'un élément donné d'un tableau s'écrira comme suit :

écrire notes(3) permet d'écrire le 3ème élément du tableau notes

Pour écrire tous les éléments du tableau Prix par exemple, on utilisera également une boucle comme suit :

```
Pour I allant de 1 à 5 faire
                    écrire Prix( I )
```

6.2.3 Exercice

Exercice1 :

Soit un vecteur qui contient l'ensemble des notes des élèves, on veut calculer la moyenne.

6.3 Méthodes de tri d'un vecteur

Un tableau est trié :

- Par ordre croissant, si le contenu d'une case d'indice i est inférieur ou égal au contenu de la case d'indice i+1.
- Par ordre décroissant, si le contenu d'une case d'indice i est supérieur ou égal au contenu de la case d'indice i+1.

6.3.1 Tri croissant

Le tri d'un tableau par ordre croissant se fait comme suit :

- Rechercher le minimum dans le tableau et l'amener à la case d'indice 1 par des permutations.
- Appliquer le même principe pour amener le minimum des cases restantes à la case d'indice 2, et ainsi de suite jusqu'à traiter tout le tableau.

Algorithme

WWW.DZPEN.COM | قلم الجزائر

Voici l'algorithme qui fait le tri d'un tableau par ordre croissant.

Variables i , j , intermédiaire : entier

Tableau valeurs (10) : entier

Debut

Pour i **allant de** 1 **à** 10

écrire « Entrer un nombre : »

Lire valeurs (i)

Pour i **allant de** 1 **à** 9

Pour j **allant de** i+1 **à** 10

Si valeurs (j) < valeurs (i) **alors**

Intermédiaire ← valeurs (j)

Valeurs (j) ← valeurs (i)

Valeurs (i) ← intermédiaire

Permutation

écrire « Voici le tableau trié par ordre croissant : »

Pour i **allant de** 1 **à** 10

écrire valeurs (i)

fin

Illustration :

- On déclare trois variables entières ainsi qu'un tableau de 10 éléments.

Pour notre illustration, on utilisera un tableau de cinq éléments seulement.

- Dans la 1ère boucle « Pour i », on saisit les éléments du tableau, valeurs à trier par ordre croissant :

8	5	7	1	3

- On passe ensuite à la boucle imbriquée « Pour i » « Pour j ».

WWW.DZPEN.COM | قلم الجزائر

- On entre dans la boucle « Pour i » et on commence à i égal 1.
- On entre ensuite dans la boucle « Pour j » et on commence à j égal à i + 1, c'est-à-dire 2.
- On effectue le test **SI :** Est-ce que valeurs (j), c'est-à-dire valeurs (2), égale ici à 5, est inférieure à valeurs (i),égale ici à 8 ?

  Oui. Donc, on permute valeurs (1) et valeurs (2).

Le tableau valeurs se présente donc comme suit au 1er passage dans la boucle j :

5	8	7	1	3

- On revient dans la boucle « Pour j » et on continue à la valeur de j suivante, c'est-à-dire 3.
- On effectue le test **SI :** Est-ce que valeurs (j), c'est-à-dire valeurs (3), égale ici à 7, est inférieure à valeurs (i), c'est-à-dire valeur (1), égale maintenant à 5 ?

  Non. Donc, on ne permute pas.
- On revient dans la boucle « Pour j » et on continue à la valeur de j suivante, c'est-à-dire 4.
- On effectue le test **SI :** Est-ce que valeurs (j), c'est-à-dire valeurs (4), égale ici à 1, est inférieure à valeurs (i), c'est-à-dire valeur (1), égale à 5 ?

  Oui. Donc, on permute valeurs (1) et valeurs (4).

  Le tableau valeurs se présente donc comme suit au 3ème passage dans la boucle j :

1	8	7	5	3

- On revient dans la boucle « Pour j » et on continue à la valeur de j suivante, c'est-à-dire 5.
- On effectue le test **SI :** Est-ce que valeurs (j), c'est-à-dire valeurs (5), égale ici à 3, est inférieure à valeurs (i), c'est-à-dire valeur (1), égale maintenant à 1 ?

WWW.DZPEN.COM | قلم الجزائر

Non. Donc, on ne permute pas.

A la fin de la boucle j, on voit bien que le plus petit élément du tableau se retrouve bien à la case 1 du tableau valeurs. Maintenant, il faut donc refaire la même chose avec la partie de tableau qui commence à l'élément 2.

- On revient dans la boucle « Pour i » et on contenue à la valeur suivante de i, c'est-à-dire 2.
- On entre ensuite dans la boucle « Pour j » et on commence à j égal à i + 1, c'est-à-dire 3, et ainsi de suite. A la fin des deux boucles i et j, le tableau valeurs sera trié par ordre croissant.

6.3.2 Tri par sélection

Le principe consiste à fixer le premier élément du vecteur comme minimum, puis le comparer aux autres éléments du vecteur, s'il existe un élément inférieur au minimum fixe, alors on fait la permutation, l'élément en question prend la place du minimum et on recommence jusqu'à arriver au dernier élément du vecteur.

1. Min

3	5	1	2

2. Min

3	5	1	2

3. Min

1	5	3	2

4. Min

1	5	3	2

5. Min

1	3	5	2

6. Min

1	2	5	3

7.

1	**2**	5	3

WWW.DZPEN.COM | قلم الجزائر

8.

	Min		
1	**2**	**3**	5

6.3.3 Exercice

Exercice1

On vous a expliqué le déroulement d'un tri par sélection d'un tableau, écrire l'algorithme qui permet de faire le tri par sélection d'un tableau.

6.4 Résumé

Au cours de ce chapitre, vous avez vu comment on peut représenter des données à l'aide d'un tableau à une dimension, qui peut prendre plusieurs valeurs en même temps. Un tableau regroupe des éléments de même type.

La **première section** vous a présenté la définition d'un tableau à une dimension. Les données du tableau sont aussi appelées les éléments du tableau. Chaque élément, comme une variable simple, possède une valeur unique à un instant de l'exécution. Pour les utiliser il est nécessaire de connaître la dimension (ou la taille) du tableau et le type des éléments.

A **la deuxième section**, portant sur la manipulation des tableaux a une dimension, vous avez découvert **la lecture** d'un tableau qui permet d'initialiser tous les éléments d'un tableau par des valeurs saisies par l'utilisateur, **l'affectation** pour assigner directement une valeur à un élément d'un tableau, **l'écriture** d'un tableau qui permet d'afficher les éléments d'un tableau.

La troisième section, quant à elle, vous a exposé, les deux manières de trier un tableau à une dimension.

- Par ordre croissant c'est de la plus petite valeur jusqu'à la plus grande, ou par ordre décroissant, le tri du tableau se fera inversement c'est-à-dire du maximum au minimum.
- Tri par sélection : if faut fixer le premier élément du vecteur comme minimum, puis le comparer aux autres éléments du vecteur.

WWW.DZPEN.COM | قلم الجزائر

6.5 Exercice de synthèse :

Soit V1 et V2 deux vecteurs de même taille, contenant des entiers, on veut calculer la somme des éléments des deux vecteurs et mettre le résultat dans le vecteur V3.

V1	1	13	0	22	-8
V2	25	10	-2	14	33
V3	26	23	-2	36	25

WWW.DZPEN.COM | قلم الجزائر

Chapitre 7 :

Ecriture d'un algorithme qui utilise des tableaux à deux dimensions

Lorsqu'un traitement utilise plusieurs tableaux à une dimension et qui subissent le même traitement on utilise un seul tableau de deux dimensions.

Ce tableau avec sa nouvelle forme possède un identifiant unique chaque élément est identifié par deux indices, l'un indique la ligne et l'autre la colonne.

Les tableaux à 2 dimensions sont utilisés généralement dans le calcul des matrices.

7.1 Déclaration d'une matrice

7.1.1 Définition et déclaration

Définition :

Un tableau à deux dimensions, comme un tableau à une dimension, est un ensemble de variables de même type (numérique, caractère, . . .) ayant toutes le même nom. Il est dit à deux dimensions car il se présente sous forme d'un ensemble de lignes et de colonnes.

Exemple : Le tableau **notes** ci-dessous possède 5 lignes et 7 colonnes.

notes

On a répondu précédemment à la question de l'unité d'avoir un ensemble de variables avec le même nom.

Mais comment distinguera-t-on les différents éléments d'un tableau à deux dimensions ? La réponse est simple : chaque élément est référencé par l'indice de la ligne et l'indice de la colonne sur lesquelles il se trouve. Par exemple, **notes(2,5)** désigne l'élément de la ligne 2, colonne 5.

Exemple : Soit à représenter l'ensemble des notes des élèves pour les modules d'algorithmique, d'analyse et de mathématiques :

WWW.DZPEN.COM | قلم الجزائر

	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11
Algo (M1)	10	11	12	2	14	8	6	5	6	10	12
Analyse (M2)	16	14	5	8	6	11	15	13	8	5	9
Maths (M3)	10	12	15	3	8	4	7	10	12	14	8

Ce tableau présente deux entrées : L'entrée Ei pour les élèves ;

L'entrée Mi pour les modules ;

Déclaration d'un tableau à deux dimensions (matrice):

Nous déclarons les tableaux à deux dimensions comme suit :

Tableau <nom du tableau> (nombre lignes, nombre colonnes) : **type**

Exemple 1 :

Tableau notes (10, 15) : **entier**

Dans cet exemple, nous avons déclaré un tableau de type entier, nommé notes, et possédant 10 lignes et colonnes. Ce tableau pourra contenir (10 X 15) valeurs entières, soit 150 valeurs : notes (1 , 1), notes (1 , 2), . . . , notes (1 , 15), notes (2 , 1), notes (2 , 2) , ... , notes (2 , 15) , . . . , notes (10 , 1), notes (10 , 2), . . . , notes (10 , 15).

L'utilité d'un tableau à deux dimensions réside dans la possibilité de déclarer un seul tableau au lieu d'en déclarer plusieurs identiques. En effet, le tableau **notes** ci-dessus équivaut à 10 tableaux simples de 15 éléments chacun. En d'autres termes, la déclaration :

Variables

Tableau notes (10, 15) : **entier**

Remplace : **Tableau** notes 1 (15) , **. . . ,** notes 10 (15) : **entier**

Nous utiliserons ici les mêmes conventions d'écriture que celles que nous avons utilisées pour les tableaux à une dimension. Nous écrirons par exemple :

WWW.DZPEN.COM | قلم الجزائر

Variables

Tableau X (25, 10), A (9, 50) : **entier**

Pour déclarer deux matrices de même type sur la même ligne.

7.1.2 Exercice

Exercice1 Faire la déclaration de cette matrice nommé MAT dont le nombre de lignes est de 20, le nombre de colonnes est de 6 et de type réel.

	1	**2**	. . . . .	**19**	**20**
1					
2					
.					
.					
.					
6					

Exercice2

Une famille dépense pendant les 12 mois de l'année dans la **N**ourriture, **V**êtements, **L**ogement, **D**istraction et **C**rédit. Déclarer cette matrice

Voici la matrice qui représente cette réalité.

	Janvier	Février	. . . . .	Octobre	Décembre
N					
V					
L					
D					
C					

WWW.DZPEN.COM | قلم الجزائر

7.2 Manipulation d'une matrice : Lecture, écriture

Une fois déclaré, un tableau à deux dimensions peut être utilisé comme un ensemble de variables simples. Comme pour un tableau à une dimension, Les manipulations de base sont la lecture et l'écriture.

7.2.1 La lecture d'une matrice

Il possible d'affecter des valeurs aux éléments d'un tableau à deux dimensions lors de l'exécution d'un programme. Les valeurs sont alors saisies par l'utilisateur à la demande du programme.

Pour initialiser tous les éléments d'un tableau à deux dimensions par des valeurs saisies par l'utilisateur, on utilisera une double boucle comme suit :

```
Pour I allant de 1 à N faire
        Pour J allant de 1 à M  faire
                Lire Mat ( I, J )
```

Avec :

Mat : la variable matrice.

I : un indice qui indique la ligne.

J : un indice qui indique la colonne.

N : le nombre de lignes.

M : le nombre de colonnes.

Exemple :

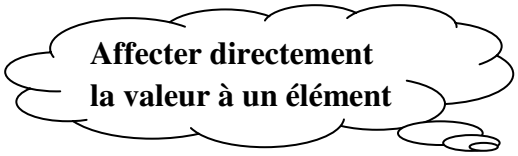
```
Variables  I , J : entier
        Tableau notes (10 , 15 ) : entier
Pour I allant de 1 à 10 faire
        Pour J allant de 1 à 15  faire
                écrire « Entrez une note : »
                Lire Mat ( I, J )
```


WWW.DZPEN.COM | قلم الجزائر

On peut aussi assigner directement une valeur à un élément d'un tableau à deux dimensions.

Exemple :

notes (1 , 5) ⟵ 15



Cette écriture signifie : assigner au l'élément de la ligne 1, colonne 5 du tableau notes la valeur 15.

- Pour initialiser les éléments d'un tableau à deux dimensions on déclare par exemple deux matrices X et Y de type entier.

Variables I , J : **entier**

Tableau X (10 , 10), Y (15 , 7) : **entier**

Pour I **allant de** 1 à 10 **faire**

Pour J **allant de** 1 à 10 **faire**

X(I , J) ⟵ 1

Pour I **allant de** 1 à 15 **faire**

Pour J **allant de** 1 à 7 **faire**

Y(I , J) ⟵ 2

1	**1**	. . . .	**1**
1	**1**	. . . .	**1**

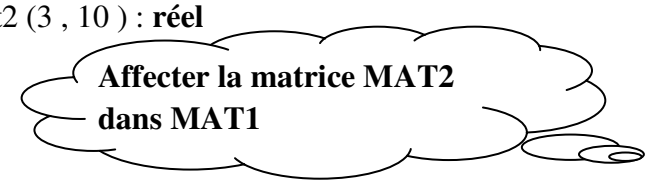
2	**2**	. . . .	**2**
2	**2**	. . . .	**2**

Dans cet exemple, on initialise tous les éléments de X a 1 et tous les éléments de Y a 2.

- Pour affecter une matrice à une autre, il faut que les deux matrices soient de même type.

Tableau Mat1 , Mat2 (3 , 10) : **réel**

Mat1 ⟵ Mat2



WWW.DZPEN.COM | قلم الجزائر

7.2.2 L'écriture d'une matrice

De façon analogue à la lecture, l'écriture d'un élément donné d'un tableau à deux dimensions s'écrira comme suit :

écrire notes(2 , 5)

L'instruction précédente permet d'écrire l'élément de la ligne 2 et la colonne 5 du tableau notes.

Pour écrire tous les éléments du tableau notes précédent, on utilisera également une double boucle comme suit :

```
Pour I allant de 1 à 20 faire
        Pour J allant de 1 à 10 faire
                écrire notes( I , J )
```

7.2.3 Exercice

Quels résultats fournira l'exécution de l'algorithme suivant :

```
Variables  I , J, somme : entier
        Tableau   T(3 , 2) : entier
Pour I allant de 1 à 3 faire
Pour J allant de 1 à 2 faire
                T(I , J)  ⟵  2 * ( I + 1) + J
Somme  ⟵  0
Pour I allant de 1 à 3 faire
Pour J allant de 1 à 2 faire
                somme  ⟵  somme + T( I , J)
écrire « Somme = », somme
```


WWW.DZPEN.COM | قلم الجزائر

7.3Résumé

Dans ce chapitre, vous avez vu le tableau à deux dimensions (matrice), comme un tableau à une dimension, est un ensemble de variables de même type ayant toutes le même nom. Il est dit à deux dimensions car il se présente sous la forme d'un ensemble de lignes et de colonnes.

La **première section**, vous a présenté la définition du tableau à deux dimensions et vous a montrer comment déclarer une matrice.

Une fois déclaré, une matrice peut être utilisée comme un ensemble de variables simples. Dans la **deuxième section**, vous avez vu les trois manipulations de base, la lecture qui permet d'initialiser tous les éléments d'une matrice par des valeurs saisies par l'utilisateur, l'affectation qui permet assigner directement une valeur à un élément d'une matrice, l'écriture qui permet d'afficher les éléments d'une matrice.

Cette section termine le chapitre et l'étude du module d'élaboration d'algorithmes simples de votre programme d'étude. Le prochain chapitre vous présentera un résumé général et une conclusion générale.

7.4 Exercice de synthèse

Ecrire l'algorithme qui calcul et affiche la table de Pythagore.

Les éléments de la matrice seront sous cette forme :

C (I , J) ⟵ I * J

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

Remarque : la matrice est carré.

WWW.DZPEN.COM | قلم الجزائر

Chapitre 8 :

Elaborer des algorithmes simples

En parcourant les sept chapitres de ce module, vous avez découvert les algorithmes et les différentes structures pour leurs élaborations.

Ce dernier chapitre fait le point sur l'ensemble des notions étudiées dans ce module. Vous y trouverez donc un résumé des principales notions exposées ainsi qu'une brève conclusion, et le corrigé des exercices, enfin une bibliographie complète le chapitre.

8.1 Exercice de synthèse global :

Ecrire l'algorithme effectuant le décalage des éléments d'un tableau.

Exemple :

- Tableau initial

D	E	C	A	L	A	G	E

- Tableau modifié (décalage à gauche) E C A L A G E D

E	C	A	L	A	G	E	D

8.2 Résumé général

Voici le résumé des sept chapitres de ce module « **élaboration des algorithme simple** ». Ce tour d'horizon vous permettra de revoir les éléments essentiels qui ont été étudiés.

Le **premier chapitre** vous a présenté la décomposition d'une action algorithmique, Vous a permis de comprendre que pour résoudre un problème donné il faudrait l'analyser et le formaliser pour qu'il puisse être automatisé. Il faut déterminer les données fournies et les résultats attendus et enfin trouver comment le résoudre. Ensuite vous avez vu la définition d'un algorithme qui est une suite d'opérations élémentaires devant être exécutées dans un ordre donné, pour accomplir une tâche donnée. Cet algorithme doit être clair et compréhensible, c'est ce qu'on a appelé qualité d'un algorithme.

Au cours du **deuxième chapitre,** vous avez exploré les données et leurs structures. Vous avez vu la représentation en mémoire des données : chaque mot de la mémoire est repéré par un numéro qui constitue son adresse. Et les adresses de la mémoire centrale sont représentées par des noms. Ce chapitre vous a présenté les constantes identifiées par un nom attribué et une valeur constante et définitive, et aussi les variables dont la valeur peut changer au cours de l'exécution. Pour chaque variable if faut spécifier le nom et le type.

WWW.DZPEN.COM | قلم الجزائر

Ce chapitre se terminait par l'étude des différents types standards qui sont le type Entier, le type Réel, le type Caractère, le type Booléen, pour chaque type on a donné la définition et la déclaration.

Vous avez commencé à manipuler des algorithmes à partir du **troisième chapitre.** Ainsi, chaque variable est représentée dans l'algorithme par son identificateur et une valeur. Cette valeur peut être placée dans la variable, nous avons appelé cette action **l'affectation.** Vous avez vu aussi une autre façon d'initialiser les variables qui est la **lecture**.

A la fin de ce chapitre vous avez trouvé l'instruction **écrire** qui vous permet d'éditer les résultats. Cependant, pour les éditer à l'écran, il est préférable d'expliciter le résultat numérique par un **message** (**commentaire**). Tous les messages encadrés par des guillemets apparaissent à l'écran tel qu'ils ont été tapés.

C'est au cours du **quatrième chapitre** que vous avez appris les notions essentielles à l'élaboration d'algorithmes complets qui peut prétendre à la résolution de n'importe quel problème. Car vous vous êtes familiarisé avec l'utilisation des structures alternatives et les structures de répétitions.

Vous avez découvert que la puissance d'un algorithme provient du fait qu'il peut effectuer des choix dans la façon d'exécuter les instructions. La forme la plus simple d'une situation de choix est celle où deux décisions sont possibles dépondant d'une condition qui doit être une expression de type booléen, ou le choix multiple qui est une généralisation de la structure alternative. Au lieu d'avoir deux possibilités, on a au moins trois, et le choix multiple associe à chaque valeur une ou plusieurs actions.

Vous avez vu également la structure répétitive, qui permet à un algorithme de répéter plusieurs fois les mêmes opérations, en utilisant la structure de répétition qui est **la boucle**. On appelle les boucles « **tant que »**, « **répéter . . . jusqu'à »,** la répétition contrôlée par des conditions, et on appelle la boucle « **Pour »** répétitions avec indice.

Le **cinquième chapitre** vous a présenté une nouvelle structure de données qui est **les chaînes de caractères,** elles vous permettent de manipuler des caractères et du texte. Une chaîne de caractère, est une suite de caractères quelconque d'une longueur maximum ne dépassant pas 255 caractères, qu'on déclare avant de l'utiliser dans un algorithme.

Par la suite, vous avez découvert les instructions et les opérations qui peuvent être effectué sur les chaînes de caractères, tels que la lecture, l'écriture, l'affectation, La comparaison, la concaténation et un ensemble de fonctions et de procédures comme la fonction « Length », la fonction d'extraction de sous–chaînes « COPY », la fonction de localisation de sous–chaînes « POS », etc.

Au cours du sixième chapitre vous avez fait connaissance avec une nouvelle structure de données qui peut prendre plusieurs valeurs en même temps à condition qu'ils soient de

WWW.DZPEN.COM | قلم الجزائر

même type, cette structure est le tableau à une dimension.

Le début du chapitre vous a présenté la définition d'un tableau à une dimension, et de ces éléments. Chaque élément, comme une variable simple, possède une valeur unique à un instant de l'exécution. Pour les utiliser il est nécessaire de connaître la dimension (ou la taille) du tableau et le type des éléments.

Vous avez appris aussi la lecture et l'écriture d'un tableau à une dimension, ainsi l'affectation pour assigner directement une valeur à un élément du tableau.

Ce chapitre se terminait sur les deux manières de trier un tableau à une dimension, la première est par ordre croissant c'est de la plus petite valeur jusqu'à la plus grande, ou par ordre décroissant, et la deuxième est le tri par sélection qui consiste à fixer le premier élément du vecteur comme minimum, puis le comparer aux autres éléments du vecteur.

Enfin, le **septième chapitre** vous a permis de connaître le tableau à deux dimensions appelé aussi matrice. C'est un ensemble de variables de même type ayant toutes le même nom. Il est dit à deux dimensions car il se présente sous la forme d'un ensemble de lignes et de colonnes.

Comme le tableau à une dimension, la matrice avant de l'utiliser dans un algorithme, doit être déclaré. Une fois déclaré, une matrice peut être utilisée comme un ensemble de variables simples.

Vous avez vu les trois manipulations de base, la lecture qui permet d'initialiser tous les éléments d'une matrice par des valeurs saisies par l'utilisateur, l'affectation qui permet assigner directement une valeur à un élément d'une matrice, l'écriture qui permet d'afficher les éléments d'une matrice.

Enfin, vous savez dorénavant comment élaborer vos algorithmes et résoudre d'une manière logique et efficace n'importe quel problème, néanmoins il reste d'autres structures plus complexes que vous développerez dans un autre module.

Vous êtes maintenant en mesure d'évaluer si vous avez bien retenu les points importants exposés dans ce module. Si vous ne comprenez pas entièrement certains éléments mentionnés dans le résumé, n'hésitez pas à reprendre la lecture de la ou des sections s'y rapportant et à refaire les exercices correspondants.

8.3 Conclusion générale

Le module « élaboration d'algorithmes simples » est maintenant terminé. Il vous a permis de découvrir les algorithmes et de les manipuler plus aisément.

Si vous avez exploré une bonne partie des instructions permettant de manipuler les algorithmes et des structures de données, d'autres existes et qui se trouvent dans un autre

WWW.DZPEN.COM | قلم الجزائر

module que vous verrez un peu plus tard dans votre formation. Néanmoins, vous avez acquis les connaissances nécessaires pour résoudre un problème donné, et vous serviront tout au long de la pratique de votre métier.

Ainsi, l'objectif de ce manuel est que vous maîtriser les outils et les techniques de programmation existantes, mais aussi vous devez être capable de résoudre, tout seul, n'importe quel type de problème.

Pour conclure, on dira que ce guide va vous permettre non seulement de maîtriser les technique de programmation, mais doit encore développer chez vous un esprit à la fois créateur et analyste.

8.3 Le corrigé des exercices

Chapitre 1 :

Exercice :

Calculer la moyenne d'une classe dans une matière :

Dans ce cas :

- Pour effectuer le calcul de la moyenne, on a besoin de la note de chaque étudiant dans la matière en question;
- On effectue ensuite le calcul de la moyenne comme suit :
  - On calcule la somme des notes ;
  - On divise par le nombre d'étudiants ;
- Enfin, on affiche la moyenne.

Exercice de synthèse

L'énoncé du problème indique :

Les données fournies : deux nombres représentant les valeurs de la somme placée et du taux d'intérêt.

Les résultats désirés : deux nombres représentant **l'intérêt fourni** par la somme placée ainsi que **la valeur obtenue** après placement d'un an.

Traitement : il nous faut maintenant décrire les différentes étapes permettant de passer des données aux résultats.

WWW.DZPEN.COM | قلم الجزائر

Pour cela il faut exprimer cette règle :

« Pour obtenir l'intérêt fourni par la somme, il suffit de multiplier la somme par le taux d'intérêt divisé par cent ; la valeur acquise s'obtient en additionnant ce dernier montant à la somme initiale.»

Chapitre 2 :

Exercice

Les mots sont représentés par des noms est repéré par un numéro qui constitue son adresse dans la mémoire centrale.

Il y'a donc deux façons de voir la mémoire centrale de l'ordinateur : côté programmeur et côté ordinateur.

Exercice

Constante prix unitaire = 5.25

Variable ligne, colonne: **numérique**

Variable voyelle: **caractère**

Exercice de synthèse

Variable note, coefficient: **entier**

Variable valeur: **réel**

Variable drapeau: **caractère**

Chapitre 3 :

Exercice

1. Entrer la longueur (variable Long)
2. Entrer la largeur (variable Larg)
3. Surface = Long x Larg
4. Afficher le résultat (Surface)

WWW.DZPEN.COM | قلم الجزائر

Exercice

Les instructions d'affectation qui permettent d'échanger les valeurs de deux variables A et B :

(1) C ⟵ A

(2) A ⟵ B

(3) B ⟵ C

	A	B	C
	5	**7**	**---**
C ⟵ A	5	7	**5**
A ⟵ B	**7**	7	5
B ⟵ C	5	**5**	5

Exercice de synthèse

L'algorithme

Debut

Variables nombre, carré, triple : **entier**

écrire « Donnez un nombre : »

Lire nombre

carré ⟵ nombre * nombre

écrire « Le carré de », nombre , « est », carré

Fin

Exécution de l'algorithme :

L'algorithme va afficher le message suivant :

Donnez un nombre :

L'utilisateur va saisir un nombre, par exemple :

5

WWW.DZPEN.COM | قلم الجزائر

L'algorithme va effectuer le calcul du carré et du triple, puis va effectuer les résultats suivants :

Le carré de 5 est 25

Chapitre 4 :

Exercice (les conditions)

Exercice 1

L'algorithme

Variables prix, montant, remise : **réel**

quantité : **entier**

debut

écrire « Donnez le prix et la quantité : »

lire prix, quantité

montant ← prix * quantité

si montant > 500 **alors**

remise ← montant * 0.03

montant ← montant - remise

écrire « Montant à payer : », montant

Fin

Exercice 2

L'algorithme

Variables A, B, C : **entier**

Debut

écrire « Donnez le premier nombre : »

lire A

WWW.DZPEN.COM | قلم الجزائر

écrire « Donnez le deuxième nombre : »

lire B

écrire « Donnez le troisième nombre : »

lire C

Si (A > B) et (A > C) **alors**

écrire « Le maximum est », A

Sinon

Si B > C **alors**

écrire « Le maximum est », B

Sinon

écrire « Le maximum est », C

Fin

Exercice 3

1. **L'algorithme en utilisant la structure alternative Si …... alors.**

Variables C : **caractère**

Debut

écrire « Donnez un caractère : »

lire C

Si (C = 'A') or (C = 'E') or(C = 'I') or(C = 'U') or(C = 'O') or (C = ' Y') **alors**

écrire C, « est une voyelle »

Sinon

écrire C, « est une consonne »

Fin

2. **L'algorithme en utilisant le choix multiple cas ….où.**

WWW.DZPEN.COM | قلم الجزائر

Variables C : **caractère**

Debut

écrire « Donnez un caractère : »

lire C

Cas C **ou**

'A' , 'E' , 'I' , 'U' , 'O' , 'Y' **: écrire** C, « est une voyelle »

Sinon

écrire C, « est une consonne »

Fin

Fin

Exercices (les répétition)

Exercice 1

L'algorithme

Variables C : **caractère**

Debut

Pour I allant de 'A' à 'Z' **faire**

écrire C

écrire « * »

fpour

Fin

Exercice 2

L'algorithme

Variables S, I , N : **entier**

Debut

WWW.DZPEN.COM | قلم الجزائر

écrire « Donnez un nombre : »

lire N

S ⟵ 2

Pour I allant de 2 à N **faire**

S ⟵ S * I

écrire « Le factoriel de », N , « est : », S

Fin

Exercice de synthèse

L'algorithme

Variables N : **entier**

Bool : **booléen**

Debut

Bool ⟵ false

Répéter

écrire « Donnez un entier : »

lire N

Si N < 0 **alors**

écrire N

Bool ⟵ faux

Sinon

Bool ⟵ vrai

Jusqu'à

Bool ⟵ vrai

Fin

WWW.DZPEN.COM | قلم الجزائر

Chapitre 5 :

Exercice (Déclaration)

Exercice 1

Déclaration

Variables

Chaine phrase (20)

Exercice 2

Les opérations sur les chaînes de caractères sont :

- Déclaration des variables de chaîne de caractère de longueur variable.
- La lecture et l'écriture de chaîne.
- L'affectation entre chaîne.
- Existence d'opérateurs de comparaison et de concaténation.
- Un jeu complet de fonctions et de procédures portant sur les chaînes de caractères.

Exercice (les opérations sur les chaînes)

Exercice 1

L'algorithme

Variables **Chaine** CH (80)

J , e , u , a , i , o : **entire**

Debut

e ← 0 u ← 0 a ← 0 i ←0 o ← 0

écrire « Donnez le texte : »

lire CH

Pour J allant de 1 à Length (CH) **faire**

Cas CH (J) **ou**

WWW.DZPEN.COM | قلم الجزائر

‘e’, ‘ E’ **:** e ⟵ e + 1

‘o’, ‘O’ **:** o ⟵ o + 1

‘u’, ’U’ **:** u ⟵ u + 1

‘i’ , ‘I’ **:** i ⟵ i + 1

‘a’ ,‘A’**:** a ⟵ a + 1

Fin

écrire « Le nombre de e : » , e

écrire « Le nombre de o : » , o

écrire « Le nombre de u : » , u

écrire « Le nombre de i : » , i

écrire « Le nombre de a : » , a

Fin

Exercice 2

L’algorithme

Variables Chaine CH (7)

N , I : **entier**

Debut

Répéter

écrire « Donnez un entier : »

lire N

STR (N , CH)

Jusqu’à length (CH) < 7

Pour I allant de 1 à 7 - length (CH) **faire**

CH ⟵ ‘*’ + CH

écrire « La chaine résultat est : » , CH

WWW.DZPEN.COM | قلم الجزائر

Fin

Exercices de synthèse

L'algorithme

Variables **Chaine** PH (20)

S , J , I : **entier**

Debut

écrire « Donnez la chaîne : »

lire PH

S ⟵ Length (PH)

J ⟵ S

Pour I allant de 1 à S **faire**

écrire copy (PH , 1 , I) + copy (PH , J , I)

J ⟵ J - 1

Fin

Chapitre 6 :

Exercice (Déclaration)

Exercice 1

Déclaration

Variables

Tableau VECT (20) : **réel**

Exercice 2

Créons un tableau contenant le nombre de jour de chacun des moins d'une année non bissextile.

Janvier	février	mars	avril	mai	juin	juillet	août	septembre	octobre	novembre	décembre
31	28	31	30	31	30	31	31	30	31	30	31

WWW.DZPEN.COM | قلم الجزائر

Ce tableau est composé de douze données, attribuons-lui l'identificateur **TABMOIS**.

Il est schématisé comme suit :

31	**28**	**31**	**30**	**31**	**30**	**31**	**31**	**30**	**31**	**30**	**31**

Exercice

Exercice 1

L'algorithme

Variables

Tableau notes (20) : **réel**

I , N : **entier**

Somme, Moyenne : **réel**

Debut

écrire « Donnez le nombre de notes : »

lire N

I ⟵ 1

Tant que I < = N **faire**

écrire « Note (», I , «) = »

lire notes (I)

Si (notes (I) > 20) ou (notes (I) < 0) **alors**

écrire « La note hors de l'intervalle 0 . . 20 »

Sinon

I ⟵ I + 1

Somme ⟵ 0

Pour I allant de N **faire**

Somme ⟵ Somme + notes (I)

WWW.DZPEN.COM | قلم الجزائر

Moyenne ⟵ Somme / N

écrire « La moyenne est : », Moyenne

Fin

Exercice (tri d'un tableau)

Exercice 1

L'algorithme

Variables

Tableau Vect (20) : **entier**

Min, J , I , N , X : **entier**

Debut

écrire « Donnez le nombre des éléments du vecteur : »

lire N

Pour I allant de 1 à N **faire**

écrire « Vect (», I , «) = »

lire Vect (I)

Pour I allant de 1 à N **faire**

Min ⟵ I

Pour J allant de I + 1 à N **faire**

Si Vect (J) < Vect (Min) **alors**

Min⟵ J

X ⟵ Vect (Min)

Vect (Min)⟵ Vect (I)

Vect (I) ⟵ X

écrire « ********* Voici le vecteur trié ********** »

Pour I allant de 1 à N **faire**

WWW.DZPEN.COM | قلم الجزائر

écrire « Vect (», I , «) = », Vect (I)

Fin

Exercice de synthèse

L'algorithme

Variables

Tableau V1 (20) : **entier**

V2 (20) : **entier**

V3 (20) : **entier**

I , N , S : **entire**

Debut

S ⟵ 0

écrire « Donnez le nombre des éléments des vecteurs : »

lire N

Pour I allant de 1 à N **faire**

écrire « V1 (», I , «) = »

lire V1 (I)

écrire « V2 (», I , «) = »

lire V2 (I)

Pour I allant de 1 à N **faire**

S⟵ V1 (I) + V2 (I)

V3 (I) ⟵ S

Pour I allant de 1 à N **faire**

écrire « V3 (», I , «) = », V3 (I)

Fin

WWW.DZPEN.COM | قلم الجزائر

Chapitre 7 :

Exercice

Exercice 1

Déclaration

Variables

Tableau MAT (20, 6) : **réel**

Exercice 2

Une matrice des dépenses d'une famille pendant les douze moins de l'année

	1	2	3	4	5	6	7	8	9	10	11	12
nourriture												
vêtements												
logement												
distraction												
crédit												

Cette matrice a 5 lignes et 12 colonnes. Un élément est repéré par le numéro de sa ligne et le numéro de sa colonne, qui sont les deux **indices** de la matrice.

La matrice possède un identificateur : DEPENSES

L'élément noté DEPENSES (2 , 5) est celui de la 2ème ligne et de la 5ème colonne : il représente la dépense en vêtements du mois de mai.

Exercice

L'exécution de l'algorithme:

La 1ère boucle permet d'initialiser les éléments de la matrice T comme suit :

T(1 , 1) ⟵ 2 x (1 + 1) + 1 = 5

T(1 , 2) ⟵ 2 x (1 + 1) + 2 = 6

T(2 , 1) ⟵ 2 x (2 + 1) + 1 = 7

WWW.DZPEN.COM | قلم الجزائر

T(2 , 2) ⟵ 2 x (2 + 1) + 2 = 8

T(3 , 1) ⟵ 2 x (3 + 1) + 1 = 9

T(3 , 2) ⟵ 2 x (3 + 1) + 2 = 10

La 2ème boucle permet de calculer la somme des éléments de la matrice T :

Somme ⟵ 5 + 6 + 7 + 8 + 9 + 10 = 45.

A la sortie de la 2ème boucle, l'algorithme fournira donc le résultat suivant :

Somme = 45

Exercice de synthèse

L'algorithme

```
Variables  I , J, N : entier
           Tableau  MAT (50 , 50) : entier
Debut
écrire « Donnez le nombre lignes de la matrice : »
lire N
Pour I allant de 1 à N faire
Pour J allant de 1 à N faire
                    MAT (I , J) ⟵ I * J

Pour I allant de 1 à N faire
Pour J allant de 1 à N faire
                    écrire « MAT ( », I , « , » , J , « ) = », MAT ( I , J)
Fin
```


WWW.DZPEN.COM | قلم الجزائر

Exercice de synthèse global :

Algorithme

Variable

tmp: caractère

i: entier

Debut

tmp ← T[1]

Pour i ← 1 a N-1 Faire

T[i] ← T[i+1]

Ftque

T[N] ← tmp

Fin

8.4 Bibliographie

GRAINE, S & DAOUDI, F. *Les bases de l'algorithme*, Tizi-Ouzou, L'abeille, 2011, 118p.

BELAID, Mohamed chérif. *Le champion de l'algorithmique & de la programmation*, Bouira, Pages Bleues, 2002, 206p.

LIGNELET, Patrice. *Algorithmique*, Paris, Masson, 1995,158p.

RICHARD, Chantal. *Initiation à l'algorithmique*, Paris, Berlin, 1998, 222p.

http://www.multimedialab.be

http://lapoire.developpez.com/algorithmique/

http://www.pise.info/algo/introduction.htm

WWW.DZPEN.COM | قلم الجزائر