

Chapitre 6

Les Structures de Données

Leçon 1

Les Constantes Et Les Variables

Activité :

Calculer les moyennes de la matière « Physique » pour les notes suivantes :

Coeff	DC	DS	Moy	Total
3	12.5	13	12.83	38.49
3	14	11.75	12.5	37.5
3	16.25	18	17.41	52.23

→ On constate que les valeurs de Coeff sont inchangées, tandis que DC, DS et Moy prennent des valeurs différentes.

→ DC, DS, Total et Moy sont des **variables**

→ coeff est une **constante**.

→ Un algorithme est une suite d'instructions manipulant des objets (constantes, variables).

I. Les constantes

a. Définition :

Une constante est un objet ayant une valeur fixe durant l'exécution du programme. Elle est caractérisée par un nom et une valeur.

Exemple de constantes :

Pi= 3.14

G= 9.81 (calcul de poids)

b. Déclaration :

♣ **Algorithmique :**

objet	Type/Nature	Rôle
Nom_const	Constante = valeur	Constante connue
Pi	Constante = 3.14	Constante connue
Coeff	Constante = 3	Coefficient de la matière physique

♣ **Pascal :**

CONST Nom_const= valeur ;

CONST Pi= 3.14 ;

CONST Coeff= 3;

c. Application :

Présenter une analyse ainsi que le TDO d'un programme intitulé « Périmètre » qui Calcule le périmètre d'un cercle.

Analyse :

Résultat = Ecrire (p)

Traitement =

$P \leftarrow R * 2 * \text{Pi}$

Donnée : R

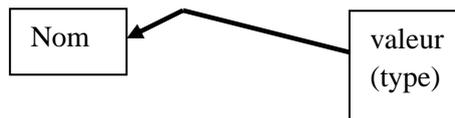
T.D.O

Objet	Type /Nature	Rôle
P	Réel	Périmètre d'un cercle
R	Réel	Rayon du cercle
Pi	Constante=3.14	Constante connue

II. Les variables :

a. Définition :

Une variable est un objet qui peut prendre différentes valeurs durant l'exécution du programme. Elle est caractérisée par un nom, un type et une valeur.



La déclaration d'une variable consiste à réserver en mémoire centrale (RAM) un espace suffisant pour la variable. Alors, la variable est un objet qui stocke les données.

L'opération qui permet de modifier le contenu d'une variable est l'affectation. Elle est représentée par « \leftarrow » au niveau algorithmique, et par « $:=$ » au niveau du Pascal.

b. Déclaration :

♣ *Algorithmique* :

objet	Type/Nature	Rôle
Nom_var	Type de la variable	--
P	réel	Périmètre d'un cercle
Moy	réel	Moyenne du physique

♣ *Pascal* :

Var Nom_var : type_var ;

Var Moy : real ;

Leçon 2 :

Les Types de Données et les Expressions

I. Les types de données :

Activité :

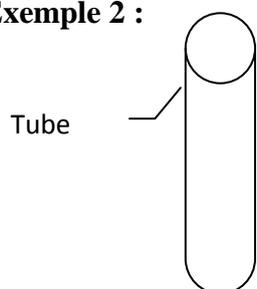
Exemple 1 :

				Type
Nom	Ayari	Manai	Sahli	Chaine de caractères
Prénom	Meriem	chiraz	Haifa	Chaine de caractères
Age	18	17	20	Entier
Moyenne Générale	13.5	14.25	15.80	Réel

Vous remarquez que chaque variable contient des valeurs de même type.

- Nom et Prénom : prennent des valeurs de type chaine de caractères.
- Age : prend des valeurs de type entier (pas de valeurs décimales)
- Moyenne Générale : prend des valeurs de type réel (il y a des valeurs décimales)

Exemple 2 :



Q : Quel est le type de matières qu'on peut les mettre dans ce tube ?

R : liquides (eau, lait, alcool, huile)

Q : Est-ce qu'on peut mettre du bois. Non.

➔ La connaissance du type d'un objet détermine l'ensemble des valeurs que peut prendre.

I. Les types de données :

Une variable est caractérisée par un nom, un type et une valeur. Le type permet de déterminer :

- L'ensemble des valeurs que peut prendre une variable.
- L'ensemble des opérations qu'on les appliquer sur les variables.
- Déterminer l'espace mémoire nécessaire au stockage de cette variable dans la mémoire.

Les types les plus connus sont :

- ✘ Le type entier : désigne les nombres entiers.
- ✘ Le type réel : désigne les nombres réels.
- ✘ Le type caractère : désigne les valeurs des caractères.
- ✘ Le type Booléen : désigne les valeurs logiques (vrai/ faux)
- ✘ Le type chaîne de caractères : désigne les valeurs des chaînes de caractères.

1. Les types numériques :

a. Le type entier :

✓ Domaine de valeurs : les entiers représentent un sous ensemble Z. Ils se limitent à l'intervalle [-32768..+32767].

Req : Toute opération en dehors de cet intervalle génère une erreur

Exemple : $x \leftarrow 40000$ } $Z \leftarrow X+Y$
 $y \leftarrow 3000$ } Dépassement de l'intervalle

- ✓ Opérateurs arithmétiques : +, -, *, Div, Mod
- ✓ Opérateurs relationnels : < >, =, <, >, >=, <=
- ✓ Déclarations en algorithmique :

objet	Type/Nature	Rôle
x	entier	-

✓ Déclarations en Pascal :

Var x: integer ;

✓ Application :

17 Mod 5 = 2
 17 DIV 5 = 3
 ((58 Div 7) Mod 2) + 5 = 5
 (49 Mod 17) div (4*3) = 1.

Application :

Est-ce qu'on peut utiliser le type entier pour représenter les valeurs suivantes :

- Nb de jours de l'année (oui)
- Nb d'élèves d'une classe (oui)
- Moyenne générale d'un élève. (non)
- Température. (oui)

b. Le type réel :

- ✓ Domaine de valeurs : Sous ensemble de l'ensemble des nombres réels R
- ✓ Exemple : 7.5, 6, 45.8, ...
- ✓ Opérateurs arithmétiques : +, -, *, /
- ✓ Opérateurs relationnels : < >, =, <, >, >=, <=
- ✓ Déclarations en algorithmique :

objet	Type/Nature	Rôle
x	réel	-

✓ Déclarations en Pascal :

Var x: real;

➤ Les fonctions prédéfinies :

Les fonctions arithmétiques standards					
Syntaxe en Algorithme	Syntaxe en pascal	Rôle de la fonction	Type de X	Type de résultat	Exemples
ABS(X)	ABS(X)	Retourne la valeur absolue de X	Entier ou Réel	Même type que X	N := ABS(-20) ; N sera égale à 20 Y := ABS(-5,8) ; Y sera égale à 5,8
ARCTAN(X)	ARCTAN(X)	Retourne la valeur en radians d'arc tangente de X.	Réel	Réel	N := ARCTAN(1) ; N sera égale à PI/4 Y := ARCTAN(0,48) ; Y sera égale à PI/6
ARRONDI(X)	ROUND(X)	Retourne l'entier le plus proche de X	Réel	Entier	N := ROUND(7,49) ; N sera égale à 7 Y := ROUND(2,5) ; Y sera égale à 3
CARRE(X)	SQR(X)	Retourne le carré de X	Entier ou Réel	Même type que X	N := SQR(4) ; N sera égale à 16 Y := SQR(2,5) ; Y sera égale à 6,25
COS(X)	COS(X)	Retourne le cosinus de X (X en radians)	Réel	Réel	N := COS(PI/2) ; N sera égale à 0 Y := COS(PI) ; Y sera égale à -1
EXP(X)	EXP(X)	Retourne l'exponentielle de X	Entier ou Réel	Même type que X	N := EXP(0) ; N sera égale à 1 Y := EXP(-0,3) ; Y sera égale à 0,5
FRAC(X)	FRAC(X)	Retourne la partie décimale fractionnaire de X	Réel	Réel	N := FRAC(-2,5) ; N sera égale à -0,5 Y := FRAC(1,82) ; Y sera égale à 0,82
INT(X)	INT(X)	Retourne la partie entière de X	Entier ou Réel	Réel	N := INT(-20) ; N sera égale à -20 Y := INT(-5,8) ; Y sera égale à -5
Ln(X)	Ln(X)	Retourne le logarithme népérien de X si X est positif sinon il provoque une erreur	Entier ou Réel	Réel	N := Ln(2) ; N sera égale à 0,69 Y := Ln(1) ; Y sera égale à 0
RACINE(X)	SQRT(X)	Retourne la racine carrée de X si X est positif sinon il provoque une erreur	Entier ou Réel	Réel	N := SQRT(9) ; N sera égale à 3 Y := SQRT(6,25) ; Y sera égale à 2,5
SIN(X)	SIN(X)	Retourne le cosinus de X (X en radians)	Entier ou Réel	Réel	N := SIN(PI/2) ; N sera égale à 1 Y := SIN(PI) ; Y sera égale à 0
TRUNC(X)	TRUNC(X)	Retourne un entier, en ignorant la partie fractionnaire décimale de X.	Réel	Entier long	N := TRUNC(-3,4) ; N sera égale à -3 Y := TRUNC(2,5); Y sera égale à 2

- ✓ Application 1: Traduire les expressions arithmétiques en algorithmique et en Pascal.

- $\sqrt{[(a-c)^2 + (b+d)^2]}$.
- $|\sqrt{(\sqrt{x+3})^3 - z^2}|$
- $7x^2 + 2x - 1$
- $-11x^3 / 7y^2$

- ✓ Application 2 :

Soit la séquence d'instructions suivantes :

A ← 3

B ← 2.5

Masse ← 12

Longueur ← 4.5

Largeur ← 1.5

Hypo ← RacineCarré (carré (A) + carré (B))

Poids ← masse * g

Surface ← longueur * largeur

1. Déclarer les variables en Pascal.
2. Traduire en Pascal les différentes affectations.

2. Le type booléen :

- ✓ Domaine de valeurs : vrai, faux.
- ✓ Déclarations en algorithmique :

objet	Type/Nature	Rôle
test	booléen	-

- ✓ Déclarations en Pascal :
Var test : boolean ;

- ✓ Opérateurs logiques sur les booléens :

Notation algo	Notation en Pascal	Rôle
Non	Not	Négation
Et	AND	Conjonction
OU	OR	Disjonction
OUex	XOR	Ou exclusif

- ✓ Table de vérité :

Expression A	Expression B	Non(A)	A ET B	A OU B	A Ouex B
F	V	V	F	V	V
F	F	V	F	F	F
V	V	F	V	V	F
V	F	F	F	V	V

✓ Application : Evaluer les propositions suivantes :

- $(10 < 5)$ Et $(8 > 0) \rightarrow$ faux
- $(\text{Round}(7.5)=7)$ OU $(\text{SQRT}(2) > 2) \rightarrow$ faux
- $(\text{Trunc}(5.26=5))$ ET $(\text{Not}(\text{ABS}(-7) = 7)) \rightarrow$ faux

✓ Priorité des opérateurs :

Opérateur	Priorité	Catégorie
Not	Haute	Op. unaire
*,/,Div,Mod, AND	2	Op. multiplicatifs
+,-,OR, XOR	3	Op. Additifs
< >, =, <, >, >=, <=	4	Op. Relationnels

✓ Application :

Soit a, b et c trois variables booléennes.

1. Donner les étapes d'évaluations de l'expression **a OU b ET c**.
2. Comment faut il faire pour évaluer **a OU b** au début ?

→ Si deux opérateurs de même ont la même priorité, le calcul de l'expression se fera de gauche à droite.

→ Les expressions entre parenthèses sont plus prioritaires.

3. Le type de caractère :

✓ Domaine de valeurs : un caractère est représenté par le caractère lui-même.
On distingue plusieurs types de caractères.

- les lettres alphabétiques en majuscule et minuscule.
- les chiffres.
- les symboles.
- les caractères nom imprimables.

Un caractère est placé entre deux apostrophes en algorithmique et Tous les caractères sont ordonnés selon leurs codes ASCII.

exp : 'A'=65
'B'= 66

✓ Déclarations en algorithmique :

objet	Type/Nature	Rôle
c	caractère	-

✓ Déclarations en Pascal :

Var c : char ;

- ✓ Les opérateurs relationnels : <, >, =, <=, >=, <=
- On peut utiliser (Dans / IN)

exp : 'A' Dans ['A'..'Z'] vaut vrai.

'A' < 'B' vaut vrai.

'g' > 'b' vaut vrai.

Quelle est la différence entre '7' et 7

* '7' est un caractère.

* 7 est un entier.

- ✓ Les fonctions prédéfinies :

Nom	Code Pascal	Rôle	Exemples
ORD(c)	ORD(c)	Renvoie le code ASCII du caractère c	Ord('A') vaut 65
CHR(n)	CHR(n)	Renvoie le caractère qui correspond au code ASCII	Chr(65) vaut 'A'
PRED(c)	PRED(c)	Renvoie le caractère prédécesseur de c	Pred('B') vaut 'A'
SUCC(c)	SUCC(c)	Renvoie le caractère successeur de c	Succ('A') vaut 'B'
MAJUS(c)	UPCASE(c)	Convertit le caractère en majuscule	Upcase('c') vaut C.

- ✓ Concaténation des caractères :

L'opérateur '+' permet la concaténation des caractères.

exp : 'a'+'b' vaut 'ab'. N.B : Le type de retour est chaîne de caractères.

'1' + '7' vaut '17' ≠ 1+7 vaut 8

- ✓ Application :

* CHR(ORD('A')+1) = 'B'

* ORD(SUCC(CHR(122))) = 123

* ORD(PRED(CHR(50))) = 49

* CHR(ORD('T')+32) = 't'.

4. Le type chaîne de caractères :

a. Définition :

Une chaîne de caractères est une suite composée de N caractères. N est compris entre 0 et 255.

Une chaîne de caractère doit être placée entre deux guillemets (") au niveau de l'algorithme, et entre deux quotes simples (').

Exemple :

ALGORITHME	PASCAL
"Bonjour"	' Bonjour '
"L'élève"	' L"élève '
" " (chaîne vide)	' ' '
" " (espace)	' ' '

b. Déclaration :

✓ Déclarations en algorithmique :

OBJET	TYPE / NATURE
Ident_chaîne	CHAINE [longueur]
CH1	Chaîne [10]
CH2	Chaîne

✓ Déclarations en Pascal :

```
VAR
    CH1 : STRING [10] ;
    CH2 : STRING ;
```

c. Accès aux éléments d'une chaîne de caractères :

Pour accéder à l'*i*^{ème} élément d'une chaîne, il suffit de donner l'identificateur de la chaîne suivi de l'indice *i* entre 2 crochets.

Exemple :

```
NOM ← "AMIR" ;          NOM [2] vaut "M"
                        NOM [4] vaut 'I'
```

d. Les opérateurs relationnels sur les chaînes de caractères :

- ✓ <>, =, <, >, >=, <=
- ✓ La comparaison entre les chaînes de caractères se fait caractère par caractère selon leur code ASCII.

Exemple : 'FAMILLE' < 'famille' vaut vrai.

'1999' < '99' vaut vrai.

'Bac' < 'bac' vaut vrai.

e. Les fonctions et les procédures prédéfinies :

Les fonctions standards sur les chaînes :

Algorithme	Pascal	Rôle	Exemples
Concat(ch1,ch2,..., chn);	Concat(ch1,ch2,..., chn);	Fonction qui retourne la concaténation des chaînes ch1, ch2, ..., chn.	J ← "01" M ← "01" A ← "2002" Date ← concat (J, "/", M, "/", A) donne "01/01/2002"
Long(ch)	length (ch);	Fonction qui retourne un entier représentant la longueur en caractères de la chaîne ch.	Taille1 ← long ("bonjour") donne 7 Taille2 ← long (" bonjour") donne 8
Sous_Chaine (ch, p, nbc);	copy (ch, p, nbc);	Fonction qui retourne une sous-chaîne d'une longueur nbc à partir de la position p dans ch	mot ← " INFORMATIQUE" Abreger1 ← copy (mot, 1, 5) donne " INFOR" Abreger2 ← copy (mot, 2, 5) donne "NFORM"
position (ch1, ch2)	pos (ch1, ch2)	Fonction qui retourne la position de la chaîne ch1 dans la chaîne ch2 ; si ch1 n'est pas dans ch2 elle retourne 0.	mot1 ← "INFORMATIQUE" mot2 ← "INFO" x ← pos (mot2, mot1) x prend la valeur 1 y ← pos ("MATIQUE", mot1) y prend la valeur 6 Z ← pos ("Info", mot1) z prend la valeur 0

Les procédures standard sur les chaînes :

Algorithme	Pascal	Rôle	Exemples
Efface (ch, p, n);	DELETE (ch, p, n);	Procédure qui enlève n caractères de ch à partir de la position p.	mot1 ← "INFORMATIQUE" mot2 ← "" mot2 ← efface (mot1, 5, 8) mot2 devient "INFO"
INSERT (ch1, ch2, p);	INSERT (ch1, ch2, p);	Procédure d'insertion de la chaîne ch1 dans la chaîne ch2 à partir de la position p.	ch1 ← "ROM" ch2 ← "CD" Insert (ch1, ch2, 4) ch2 devient "CD ROM"
Convch (d, ch1);	STR (d, ch1);	Procédure qui convertit un nombre décimal d en chaîne de caractère et l'affecte à la variable ch1.	Convch (2002, ch) ch contient "2002"
Valeur (ch, d, erreur);	VAL (ch, d, erreur);	Procédure qui convertit une chaîne ch en une valeur numérique décimale et l'affecte à la variable d.	Ch1 ← "12.8" VAL (ch1, d, e) d contient le nombre 12.8 et e contient 0 (aucune erreur) ch2 ← "01/01/2002" VAL (ch2, d, e) d contient le nombre 0 et e contient 3 (erreur à la position 3 due au caractère "/" qui n'est pas un chiffre).

- ✘ Application : Donner le résultat des n
 $Ch := 'Works$ est un logiciel intégré'.
 $x := \text{Lenght}(ch) \rightarrow x := 29$
 $y := \text{copy}(ch, \text{length}(ch)-6, 3) \rightarrow y := 'int'$
 $p := \text{pos}('est', ch) \rightarrow p := 7$
 $ch[2] := \text{upcase}(ch[2]). \quad ch[2] := 'O'$.

5. Autres types:

a. Le type scalaire énuméré :

1. Définition :

- ✘ Le type scalaire énuméré définit un ensemble **fini et ordonné** de valeurs désignées par des identificateurs.

Exemple : Définir un type comportant 4 matières selon votre choix.

Matière = (Math, Physique, Informatique, Sciences).

- ✘ Un variable de type scalaire énuméré doit prendre une seule valeur parmi celles présentées par le type.

Exemple : $\text{var_mat} \leftarrow \text{Math}$.

2. Les fonctions prédéfinies sur les types énumérés :

Les valeurs d'un type énuméré sont ordonnées selon leurs ordres de déclaration.

Matière = (Math, Physique, Informatique, Sciences).

(0) (1) (2) (3)

Donc, on peut déterminer le N° d'ordre d'une valeur à travers la fonction **Ord**.

- ✘ **Ord (expression)** : donne le N° d'ordre de l'expression.

Exemple : $\text{Ord}(\text{Physique}) = 1$

$\text{Ord}(\text{Sciences}) = 3$.

- ✘ **Pred (expression)** : donne le prédécesseur d'une expression

Exemple : $\text{Pred}(\text{Math}) = \text{erreur}$.

$\text{Pred}(\text{Informatique}) = \text{Physique}$.

- ✘ **Succ (expression)** : donne le successeur de l'expression.

Exemple : $\text{Succ}(\text{Sciences}) = \text{erreur}$.

$\text{Succ}(\text{Informatique}) = \text{Sciences}$.



Remarque :

L'utilisation des valeurs appartenant aux types prédéfinies n'est pas possible.

Exemple :

$\text{Impair} = (1,3,5,7) \rightarrow$ Cette déclaration n'est pas possible car 1,3,5,7 sont des entiers.

3. Déclaration :

Algorithmique

Tableau de déclaration des nouveaux types

T.D.N.T

Type
Nom_type =(val1, val2,, valn)
Couleur = (bleu, rouge, vert)

T.D.O

objet	Type/Nature	Rôle
col	couleur	-

Pascal

Type

Nom_type =(val1, val2,, valn) ;

Couleur = (bleu, rouge, vert) ;

Var

Col : Couleur ;

→ Ou bien

Var

Col : (bleu, rouge, vert) ;

b. Le type intervalle :

1. Définition :

Un type intervalle possède les mêmes propriétés qu'un type scalaire ordonné (caractère, entier).

La définition de l'intervalle est basée sur les deux constantes **Borneinf** et **Bornesup** où (Binf < Bsup).

Exemple :

- ✗ Indice = 1..15
- ✗ Saison = (automne, hivers, printemps, été)
- Année_scolaire = automne..Printemps.

Une variable de type intervalle doit prendre obligatoirement une valeur de l'intervalle.

Algorithmique

Tableau de déclaration des nouveaux types

T.D.N.T

Type
Nom_type = Binf..Bsup
Mois = 1..12

T.D.O

objet	Type/Nature	Rôle
V_mois	Mois	-

Pascal

Type

Nom_type = Binf..Bsup ;

Mois = 1..12;

Var

V_mois : Mois ;

→ Ou bien

Var

V_mois : 1..12 ;

II. Les expressions :

Activité :

Donner le type et la valeur de chacune de ces expressions.

1. Round (7.5) <= 7 →Faux
 2. SQRT (25) +3 → 8
 - La 1^{ère} expression est logique, le résultat de type booléen.
 - La 2^{ème} expression est arithmétique, le résultat de type entier.
- Il y a deux types d'expressions

1. Définition :

Une expression est une composition d'opérateurs et d'opérandes réalisant un calcul déterminé. Il y a deux types d'expressions : les expressions arithmétiques et les expressions logiques.

2. Les opérandes :

Se sont les éléments sur lesquels on applique l'opération.

Ils peuvent être des valeurs (constante ou variables) ou des valeurs renvoyées par par des fonctions.

3. Les opérateurs :

3.1. Les opérateurs arithmétiques :

a) Les opérateurs unaires : Un opérateur unaire est appliqué à un seul opérande.

opérateur	Type opérande	Type résultat
-	Entier/réel	Entier/réel

Exemple : (-5)

b) Les opérateurs binaires : Un opérateur binaire est appliqué à deux opérandes.

	opérateur	Type opérande 1	Type opérande 2	Type résultat
Opérateurs multiplicatifs	*	Entier/réel	Entier/réel	Entier/réel
	/	réel	Entier/réel	Entier/réel
	DIV, MOD	Entier	Entier	Entier
Opérateurs additifs	+,-	Entier/réel	Entier/réel	Entier/réel

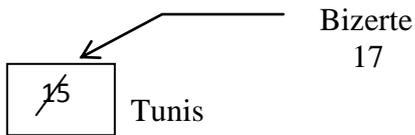
Leçon 3 :

Les Tableaux

Activité :

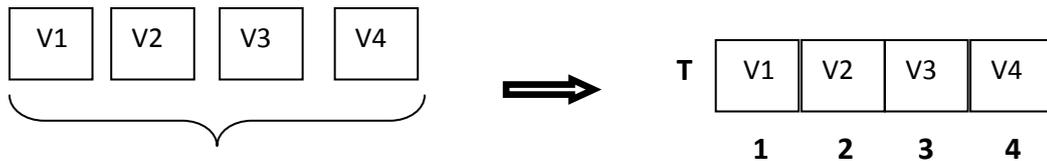
1. Le centre de météo désire enregistrer la pluviométrie mesurée dans les différentes villes de la Tunisie (24 villes). Qu'est ce qu'on va utiliser ?

⇒ L'utilisation d'une seule variable n'est pas possible.



⇒ On va utiliser pour chaque ville une variable. Mais la manipulation de plusieurs variables n'est pas facile.

⇒ Pour faciliter la manipulation de plusieurs variables, on va les regrouper dans une seule Structure appelée Tableau.



⇒ Quel est le type de données enregistrées dans le tableau ?

→ Réel

⇒ Combien de cases le tableau doit comporter ?

→ 24.

⇒ On constate qu'un tableau est une structure de données caractérisée par un nom (T), une taille (24) et un type de données (Réel).

Exemple de tableau :

Le registre d'appel est un tableau qui contient les noms des élèves.

I. définition :

Un tableau est une structure de données comportant un nombre fini de valeurs de même type.

Un tableau est caractérisé par :

- ✗ Un nom
- ✗ Une dimension
- ✗ Un type de données

T1	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 2px 10px;">12</td> <td style="padding: 2px 10px;">14</td> <td style="padding: 2px 10px;">5</td> <td style="padding: 2px 10px;">6</td> <td style="padding: 2px 10px;">30</td> </tr> <tr> <td style="padding: 2px 10px;">"a"</td> <td style="padding: 2px 10px;">"b"</td> <td style="padding: 2px 10px;">"c"</td> <td style="padding: 2px 10px;">"d"</td> <td style="padding: 2px 10px;">"e"</td> </tr> </table>	12	14	5	6	30	"a"	"b"	"c"	"d"	"e"	Tableau de 5 entiers
12	14	5	6	30								
"a"	"b"	"c"	"d"	"e"								
T2	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 2px 10px;">12.5</td> <td style="padding: 2px 10px;">0.25</td> <td style="padding: 2px 10px;">3.2</td> <td style="padding: 2px 10px;">6</td> <td style="padding: 2px 10px;">-7.1</td> </tr> <tr> <td style="padding: 2px 10px;">-1</td> <td style="padding: 2px 10px;">0</td> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">3</td> </tr> </table>	12.5	0.25	3.2	6	-7.1	-1	0	1	2	3	Tableau de 5 réels
12.5	0.25	3.2	6	-7.1								
-1	0	1	2	3								
T3	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 2px 10px;">"a"</td> <td style="padding: 2px 10px;">"m"</td> <td style="padding: 2px 10px;">"i"</td> <td style="padding: 2px 10px;">"z"</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">4</td> </tr> </table>	"a"	"m"	"i"	"z"	1	2	3	4	Tableau de 4 caractères		
"a"	"m"	"i"	"z"									
1	2	3	4									
T4	<table border="1" style="border-collapse: collapse; width: 100%;"> <tr> <td style="padding: 2px 10px;">"donia"</td> <td style="padding: 2px 10px;">"anis"</td> <td style="padding: 2px 10px;">"rim"</td> <td style="padding: 2px 10px;">"ahlem"</td> </tr> <tr> <td style="padding: 2px 10px;">1</td> <td style="padding: 2px 10px;">2</td> <td style="padding: 2px 10px;">3</td> <td style="padding: 2px 10px;">4</td> </tr> </table>	"donia"	"anis"	"rim"	"ahlem"	1	2	3	4	Tableau de 4 chaînes		
"donia"	"anis"	"rim"	"ahlem"									
1	2	3	4									

II. L'accès à un élément du tableau :

L'accès à un élément du tableau se fait par le nom du tableau et l'indice de l'élément.

Exemple :

T1 ["b"] = 14 T3 [4] = "z"
 T2 [-1] = 12.5 T4 [1] = "donia".

III. La déclaration :

Première formulation :

Algorithmique

T.D.O

objet	Type/Nature	Rôle
Nom_var	Tableau de <i>taille_max</i> de <i>type_éléments</i>	-

Exemple :

objet	Type/Nature	Rôle
T1	Tableau de 5 entiers	-
T3	Tableau de 4 caractères	

Pascal

Var

Nom_var : array [**Binf..Bsup**] of *type_éléments* ;

Exemple:

T1: array ['a'..'e'] of integer;

T3: array [1..4] of char;

Deuxième formulation : On peut déclarer un type Tableau

Algorithmique

Tableau de déclaration des nouveaux types (T.D.N.T)

Type
Nom_type = Tableau de <i>taille_max</i> de <i>type_éléments</i>

T.D.O

objet	Type/Nature	Rôle
Nom_var	Nom_type	-

Exemple :

T.D.N.T

Type
Tab = Tableau de 5 entiers
Tabcar = Tableau de 4 caractères

T.D.O

objet	Type/Nature	Rôle
T1	Tab	-
T3	Tabcar	-

Application :

Faire exercice 3 et 4 page 136.

Pascal

Type

Nom_type = array [Binf..Bsup] of *type_éléments* ;

Var

Nom_var : Nom_type ;

Exemple:

Type

Tab = array ['a'..'e'] of integer;

Tabcar =array [1..4] of char;

Var

T1 : Tab ;

T3 : Tabcar ;