

CHAPITRE

Chapitre I

Les structures de données

Objectifs :

- Montrer l'importance de la notion de variable
- Identifier et manipuler les différents types de données

Plan du chapitre :

Leçon 1 :
Les variables et les types standard de données

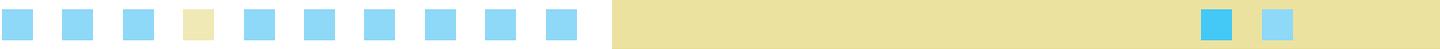
Leçon 2 :
Les expressions

Leçon 3 :
Le type scalaire énuméré et le type intervalle

Leçon 4 :
Les tableaux à une dimension

Nous rappelons dans ce chapitre des notions fondamentales telles que la constante, la variable, les types de données, les différents types d'expression et les tableaux. Toutes ces entités sont manipulées dans les algorithmes. Un algorithme est une suite finie d'instructions permettant de résoudre un problème.





Leçon 1

Les variables et les types standard de données

Objectifs spécifiques :

- Identifier et manipuler les constantes et les variables.
- Manipuler les types standard.
- Manipuler les procédures et les fonctions prédéfinies.

Plan de la leçon :

I. Les constantes et les variables

II. Les types de données

- II.1. Les types numériques
- II.2. Le type booléen
- II.3. Le type caractère
- II.4. Le type chaîne de caractères

Retenons

Exercices

Leçon 1

Les variables et les types standard de données

C'est une simplification en présence de l'objet concret infiniment complexe et perpétuellement changeant, simplification qui nous est imposée soit par les nécessités de l'action soit par les exigences de l'entendement, et qui consiste à considérer un élément de l'objet comme isolé alors que rien n'est isolable, et comme constant alors que rien n'est en repos.

ALAIN

I. Les constantes et les variables

Définition

Une constante est un objet ayant une valeur fixe tout le long de l'exécution d'un algorithme ou d'un programme.

Une constante est caractérisée par :

- son nom (un identificateur unique).
- sa valeur.

Activité 1

Identifier quelques constantes utilisées dans votre cours de mathématique.

Réponse :

La constante g , utilisée dans le calcul du poids, vaut 9.81. La constante π , utilisée dans le calcul trigonométrique, vaut 3.141. Ces constantes seront déclarées dans le langage PASCAL de la manière suivante :

```
CONST  g = 9.81 ;
        pi = 3.141 ;
```

Définition

On appelle variable un objet pouvant prendre différentes valeurs tout le long de l'exécution d'un algorithme ou d'un programme.

Une variable est caractérisée par :

- son nom (un identificateur unique)
- son type
- son contenu.

Remarques :

- 1- Le choix de l'identificateur d'un objet doit être fait de manière à être le plus significatif possible.
- 2- Lorsqu'on déclare une variable, on réserve en mémoire vive (RAM) un espace mémoire propre à la variable. En effet, la variable constitue le moyen de stocker les données. C'est pour cette raison que la notion de variable est une notion fondamentale en algorithmique. Généralement, l'opération permettant de changer le contenu d'une variable est l'affectation.

Pour réussir un algorithme, il faudra d'une part déclarer les différentes variables utiles et nécessaires pour le bon déroulement de l'algorithme en définissant correctement leurs types et d'autres parts leur donner les valeurs correctes tout le long de l'algorithme. Certaines variables serviront de données initiales, d'autres seront des variables intermédiaires et d'autres variables seront les résultats de l'algorithme. Bien entendu, une variable peut être une combinaison de deux ou trois classes (donnée, intermédiaire ou résultat).

Activité 2

Dans le calcul d'un salaire annuel, on pourra utiliser la constante `nb_mois` et les variables `Salaire_annuel` et `Salaire_mensuel`.

$$\text{Salaire_annuel} = \text{nb_mois} \times \text{Salaire_mensuel}$$

- 1) Elaborer le tableau de déclaration des objets intervenant dans cette affectation.
- 2) Déclarer les différents objets en Pascal.

Réponses :

1)

Tableaux de déclaration des objets

Objet	Type/nature	Rôle
...
<code>nb_mois</code>	Constante = 12	Nombre de mois de l'année
<code>Salaire_annuel</code>	Réel	Salaire annuel de l'employé
<code>Salaire_mensuel</code>	Réel	Salaire mensuel de l'employé
...

- 2) **CONST** `nb_mois=12 ;`
VAR `Salaire_annuel : REAL ;`
 `Salaire_mensuel : REAL ;`

Activité 3

Soit la séquence d'affectations suivante :

- 1) $i := 50$;
- 2) $j := 70$;
- 3) $k := i$;
- 4) $i := j$;
- 5) $j := k$;

- 1) Donner le résultat de l'exécution de cette séquence.
- 2) Quelles sont les valeurs finales de i et de j ?
- 3) Quel est le rôle de cette séquence ?
- 4) Quelle est l'utilité de la variable k ?

Réponses :

1) Nous pouvons suivre l'évolution des valeurs des variables i , j et k en les regroupant dans un tableau constituant la trace de la séquence :

Trace de la séquence			
N° de l'instruction	i	j	k
1	50	-	-
2	50	70	-
3	50	70	50
4	70	70	50
5	70	50	50

A l'instruction N°1, seule la variable i prend la valeur 50.

A l'instruction N°2, la variable j prend la valeur 70 mais la variable i est restée inchangée.

A l'instruction N°3, k a pris la valeur de i c'est à dire 50, i et j sont restées inchangées.

A l'instruction N°4, i a changé et a pris la valeur de j c'est à dire 70 et j et k sont restées inchangées.

A l'instruction N°5, j a pris la valeur de k c'est à dire 50, i et k sont restées inchangées.

- 2) Les valeurs finales de i et de j sont respectivement 70 et 50.
- 3) Cette séquence a permis la permutation des contenus des deux variables i et j .
- 4) La variable k a assuré la sauvegarde du contenu de i avant de lui affecter j par l'instruction N°4. Elle est appelée variable intermédiaire, temporaire ou auxiliaire. En effet, si on affecte à i le contenu de j sans le sauvegarder dans k , il sera perdu.

II. Les types de données

Le type d'une variable permet de déterminer le domaine des valeurs possibles que peut prendre cette variable. La connaissance du type permet également de déduire l'ensemble des opérateurs applicables sur les variables de ce type ainsi que l'espace mémoire en octets qui sera réservé à ces variables.

Un type est désigné par un identificateur (ou un nom). Les types standard sont :

- Le type **Entier**, désignant les valeurs des nombres entiers relatifs.
- Le type **Réel**, désignant les valeurs des nombres réels.
- Le type **Caractère**, désignant les "valeurs" des caractères.
- Le type **Booléen**, désignant les valeurs logiques.
- Le type **Chaîne de caractères**, désignant les "valeurs" des chaînes de caractères.

II. 1 Les types numériques

Dans la catégorie des types numériques, nous pouvons distinguer les types Entier et Réel.

II.1.1 Le type entier

Activité 4

- 1- Quel est l'ordre de grandeur de la valeur maximale d'un entier ?
- 2- Rappeler les opérateurs arithmétiques sur les entiers.
- 3- Evaluer les expressions arithmétiques suivantes :
 - a- $17 \text{ DIV } 5$
 - b- $17 \text{ MOD } 5$
 - c- $((58 \text{ DIV } 7) \text{ MOD } 2) + 5$
 - d- $(49 \text{ MOD } 17) \text{ DIV } (4 * 3)$
- 4- Est- ce qu'on pourra utiliser le type entier pour représenter les quantités suivantes?
 - a- Nombre de jours de l'année
 - b- Durée en heures d'une séance de cours ou TP
 - c- Nombre de jours du mois de février
 - d- Salaire mensuel exprimé en dinars d'un employé.
- 5- Définir l'effet de débordement pour une variable de type entier.
- 6- Déclarer trois variables entières i , j et k en Pascal.

Réponses :

- 1- Sachant que les entiers en mathématiques forment un ensemble infini nommé Z . Or, en informatique, un entier a une représentation en machine limitée à un nombre fini d'octets (généralement 2 octets). La valeur maximale d'un entier est égale à 32767.
- 2- Les opérateurs arithmétiques sont $+$, $-$, $*$, **DIV** (Donne le quotient dans la division entière) et **MOD** (Donne le reste de la division entière).

- 3-a- Le résultat de cette expression est 3
- b- Le résultat de cette expression est 2
- c- Le résultat de cette expression est 5
- d- Le résultat de cette expression est 1

- 4-a- Le nombre de jours de l'année est un nombre entier (365 ou 366).
- b- La durée en heures d'une séance est un nombre réel car la séance peut être par exemple de 1.5 heures.
- c- Le nombre de jours du mois de février est un entier.
- d- Le salaire mensuel exprimé en dinars d'un employé est un nombre réel.

- 5- Quand on manipule une variable de type entier, il faut faire attention au risque de débordement. Quand il y a débordement au-delà des valeurs Minimale et Maximale, les calculs deviennent erronés ou provoquent des erreurs d'exécution selon les langages utilisés.

- 6- **VAR i, j, k : INTEGER;**

Remarque :

Le langage Pascal a cinq types entiers prédéfinis. Chaque type a un domaine de définition spécifique.

Type	Domaine de définition	Nombre de bits
SHORTINT	-128..127	Signé 8 bits
INTEGER	-32768..32767	Signé 16 bits
LONGINT	-2147483648..2147483647	Signé 32 bits
BYTE	0..255	Non signé 8 bits
WORD	0..65535	Non signé 16 bits

II.1.2 Le type réel

Activité 5

- 1- Quel est le domaine des valeurs du type réel ?
- 2- Quels sont les opérateurs arithmétiques valides sur des variables de type réel?
- 3- Déclarer deux variables x et y de type réel en Pascal.

Réponses :

1- Ce type recouvre un sous-ensemble de l'ensemble des nombres réels IR. La définition de ce sous-ensemble est liée à la représentation en binaire des réels dans la machine. Un nombre réel peut occuper 6 octets en Pascal sur certaines machines. On peut le coder ainsi dans l'intervalle de -10^{38} à 10^{38} .

Exemples de nombres réels

0.	-55.36	3.14	60 10 ⁻⁹	1.23	-38.0	5.6 10 ⁶
----	--------	------	---------------------	------	-------	---------------------

5.6 10⁶ c'est-à-dire 5 600 000 s'écrira aussi 5.6E+6. La lettre E se lit : «dix puissance»

2- Ces opérateurs sont + , - , * et / (division réelle).

3- **VAR** x, y : **REAL**;

II.1.3 Les fonctions arithmétiques standards

Tous les langages de programmation offrent une bibliothèque de fonctions arithmétiques facilitant la réalisation de certains calculs. Dans l'activité suivante, nous étudions quelques unes d'entre elles.

Activité 6

Soient les fonctions arithmétiques suivantes :

Tronc(-8.224)	Tronc(3.141)	Tronc(334.8)
Arrondi(712.499)	Arrondi(12.50)	Arrondi(12.99)
Abs(-7)		
Carré(7)		
RacineCarré(2)		
Sin(1.5705)		
Cos(1.5705)		
Tang(3.141)		
Cotang(1.5705)		
Ent(3.7)	Ent(-5.5)	
Aléa	Aléa(7)	
Ln(1.0)		
Exp(0.0)		

Elaborer un tableau de 6 colonnes donnant le nom algorithmique de chacune des fonctions précédentes, le code en pascal, le type du paramètre (nombre entre parenthèses), le rôle de chacune des fonctions et les résultats d'évaluation.

Réponse :

Nom algorithmique	Code en Pascal	Type du paramètre x ou n	Type du Résultat	Rôle	Exemples
Tronc (x)	TRUNC (x)	Entier ou réel	Entier	supprime la partie décimale pour ne laisser que la composante entière de x.	Tronc (-8.224) vaut -8 Tronc (3.14) vaut 3 Tronc (334.8) vaut 334
Arrondi (x)	ROUND (x)	Entier ou réel	Entier	donne un entier qui est la valeur du réel x arrondie à la plus proche valeur.	Arrondi(712.499) vaut 712 Arrondi (12.50) vaut 13 Arrondi (12.99) vaut 13
Abs (x)	ABS (x)	Entier ou réel	Entier ou Réel (même Type que x)	donne la valeur absolue de x.	Abs (-7) vaut 7
Carré (x)	SQR (x)	Entier ou réel	Entier ou Réel (même Type que x)	donne le carré de x.	Carré (7) vaut 49
Racine Carré (x)	SQRT (x)	Entier ou réel	Réel	donne la racine carrée de x si x n'est pas négatif et provoque une erreur, sinon.	RacineCarré (2) vaut 1.414 ...
Sin (x)	SIN (x)	Entier ou réel	Réel	donne le sinus de x (x en radians).	Sin(1.5705) vaut 1
Cos (x)	COS (x)	Entier ou réel	Réel	donne le cosinus de x (x en radians).	Cos(1.5705) vaut 0
Tang (x)	TAN (x)	Entier ou réel	Réel	donne la tangente de x. (x en radians).	Tang(3.141) vaut 0
Cotang (x)	COTAN (x)	Entier ou réel	Réel	donne la cotangente de x. (x en radians).	Cotang(1.5705) vaut 0
ENT(x)	INT (x)	Entier ou réel	Entier	donne la partie entière d'un réel.	ENT(3.7) vaut 3 ENT(-5.5) vaut -6
Aléa	RANDOM	Entier	Entier	donne un réel compris entre 0 et 1 exclus.	Aléa pourrait produire 0.36 par exemple.
Aléa(n)	RANDOM (n)			donne un entier entre 0 et n-1	Aléa(7) pourrait produire 2 par exemple.
Ln(x)	Ln(x)	Entier ou réel	Réel	renvoie le logarithme népérien d'un réel x	Ln(1.0) vaut 0
Exp(x)	Exp(x)	Entier ou réel	Réel	renvoie l'exponentiel de x	Exponentiel de (0.0) vaut 1

Activité 7

Soit la séquence d'affectations suivante :

$a \leftarrow 3$

$b \leftarrow 2.5$

masse $\leftarrow 12$

longueur $\leftarrow 4.5$

largeur $\leftarrow 1.5$

hypo $\leftarrow \text{RacineCarré}(\text{carré}(a)+\text{carré}(b))$

poids $\leftarrow \text{masse} * g$

surface $\leftarrow \text{longueur} * \text{largeur}$

1- Déclarer en Pascal les différents objets.

2- Traduire en Pascal les différentes affectations.

Réponses :

1- **CONST** g=9.81;

VAR masse : **INTEGER**;

a,b,poids,longueur,largeur,surface,hypo : **REAL**;

2- a:=3;

b :=2.5 ;

masse :=12 ;

longueur :=4.5 ;

larguer :=1.5 ;

hypo := **SQRT(SQR(a)+SQR(b))**;

poids := masse * g ;

surface := longueur * largeur ;

II.2 Le type booléen

Le type booléen est utilisé pour caractériser des objets de type logique.

Activité 8

1- Quelles sont les valeurs du type booléen?

2- Quels sont les opérateurs logiques qu'on peut appliquer sur les booléens?

3- Evaluer les propositions logiques suivantes :

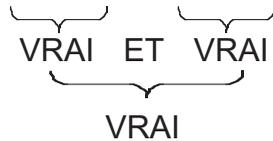
a- La proposition (88>66) ET (44<55)

b- La proposition (88>66) ET (66<55)

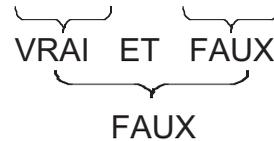
Réponses :

- 1- Les variables d'un tel type peuvent prendre uniquement deux valeurs logiques **VRAI** et **FAUX** (**TRUE** ET **FALSE** en **Pascal**).
- 2- Ces opérateurs sont NON (négation), Et (conjonction), OU (disjonction) et OUex (OU exclusif).

3- a- $(88 > 66)$ ET $(44 < 55)$



3- b- $(88 > 66)$ ET $(66 < 55)$



Activité 9

- 1- Soient a, b et c trois variables booléennes
 - a- Donner les étapes d'évaluation de l'expression a OU b ET c.
 - b- Si on veut d'abord évaluer la disjonction a OU b, comment faut-il s'y prendre ?
- 2- Dans la réponse à la question précédente, nous avons utilisé le fait que l'opérateur ET est prioritaire par rapport à l'opérateur OU. Donner l'ordre de priorité qui existe entre les opérateurs logiques.
- 3- Déclarer en Pascal quatre variables p, q, existe et drapeau de type booléen.
- 4- Evaluer les expressions logiques suivantes :
 - a- $(x \leq 4)$ ET $(x \geq 1)$ avec $x = 3$
 - b- $(x \leq 2)$ ET $(x \geq 0)$ avec $x = -4$
 - c- NON $(x \leq 55)$ OU $(x \geq 0)$ avec $x = 21$

Réponses :

- 1- a- on évalue d'abord la conjonction b ET c puis la disjonction a OU (b ET c) comme s'il y avait des parenthèses.
 b- il suffit d'ajouter des parenthèses (a OU b) ET c.
- 2- Il existe un ordre de priorité entre les opérateurs logiques :
 - La négation NON est prioritaire par rapport à la conjonction ET.
 - La conjonction ET est prioritaire par rapport à la disjonction OU.
 - La disjonction OU a la même priorité que l'opérateur OUex.
 Si deux opérateurs ont la même priorité, le calcul de l'expression logique se fera de gauche à droite.
 Dans tous les cas, les opérations mises entre parenthèses sont prioritaires.
 Les parenthèses les plus internes sont les plus prioritaires.
- 3- **VAR** p, q, existe, drapeau : **BOOLEAN** ;
- 4- a- VRAI
 b- FAUX
 c- VRAI

II.3 Le type caractère

Le type caractère est utilisé pour représenter une lettre minuscule, une lettre majuscule, un chiffre ou un signe de ponctuation, mais aussi un espace typographique, une tabulation, un retour à la ligne et quelques autres opérations spéciales (sonnerie, effacement, etc.). Tous les caractères sont ordonnés selon leur code ASCII (voir Annexe à la fin du livre).

Activité 10

- 1- Les chiffres, les lettres en majuscule, les lettres en minuscule ont des codes ordonnés et contigus ; Quel est cet ordre ?
- 2- Citer quelques opérateurs applicables aux caractères.
- 3- Déclarer en Pascal :
 - a- une constante caractère de valeur "v"
 - b- deux variables de type caractère car1 et car2

Réponses :

- 1- On peut comparer les caractères comme suit :
 "0" < "1" ... <"9" < ... "A" < "B" ... < "Z" ... "a" < "b" < "z" ...
- 2- Comme les valeurs de type caractère sont ordonnées, les opérateurs relationnels y sont définis.
Exemples :
"B" < "F" est une proposition VRAIE.
"g" > "b" est une proposition VRAIE.
- 3- a- **CONST** car = 'v' ;
 b- **VAR** car1 , car2 : **CHAR** ;".

Les fonctions prédéfinies

Parmi les fonctions prédéfinies à grand usage, nous citons :
 ORD, CHR, SUCC et PRED.

Activité 11

- 1- Soient les exemples de fonctions prédéfinies suivantes :

ORD ("A")	ORD("a")
CHR (65)	CHR (97)
SUCC ("F")	SUCC ("5")
PRED ("Z")	PRED ("9")
MAJUS("c ")	

En utilisant le tableau ASCII de l'annexe du livre, élaborer un tableau montrant le nom de la fonction, son code en Pascal, son rôle et sa valeur.

- 2- Evaluer les expressions suivantes :
- a- CHR (ORD (c)), c est une variable de type caractère.
 - b- ORD (CHR (n)), n est un entier.

- 3- Que désigne chacune des notations suivantes ?
- a- t
 - b- "t"
 - c- 7
 - d- "7"

Réponses :

- 1- Supposons que c est une variable de type caractère et que n est un entier compris entre 0 et 255.

Nom	Code en Pascal	Rôle	Exemples
ORD (c)	ORD (c)	renvoie le code ASCII du caractère c. Le résultat est un entier positif.	ORD ("A") vaut 65 ORD ("a") vaut 97
CHR (n)	CHR (n)	renvoie le caractère dont le code ASCII est n.	CHR (65) vaut "A" CHR (97) vaut "a"
SUCC (c)	SUCC (c)	renvoie le caractère successeur de c s'il existe.	SUCC ("F") vaut "G" SUCC ("5") vaut "6"
PRED (c)	PRED (c)	renvoie le caractère prédécesseur de c s'il existe.	PRED ("Z") vaut "Y" PRED ("9") vaut "8"
MAJUS (c)	UPCASE(c)	Convertit le caractère c en majuscule s'il est possible.	MAJUS("c ") vaut "C"

- 2- a- c
b- n
- 3- a- t ainsi écrit, représente un objet identifié par l'identificateur t ; c'est par exemple une variable.
b- t entre guillemets désigne le caractère t.
c- 7 désigne l'entier 7.
d- "7" désigne le caractère 7.

II.4 Le type chaîne de caractères

Une chaîne de caractères est une suite ordonnée de caractères. En algorithmique, la valeur d'une chaîne se note en utilisant des guillemets ("). En langage Pascal, on utilise plutôt des quotes simples (').

Exemple

Algorithmique	Pascal
"baccalauréat"	'baccalauréat'
"l'algorithmique"	'l'algorithmique'

Les variables chaînes de caractères sont définies par une déclaration pouvant indiquer le nombre maximum de ses caractères.

Activité 12

- 1- Elaborer un tableau de déclaration des objets où vous définirez une chaîne prenom de 10 caractères, une chaîne nom pouvant contenir jusqu'à 255 caractères et une chaîne adresse de 25 caractères.
- 2- Déclarer ces objets en Pascal.
- 3- Réaliser les affectations suivantes :
Chaîne vide à la variable Prenom, "Aloui" à la variable Nom et un espace à la variable Adresse.
- 4- Comment accéder au i ème caractère d'une chaîne CH ?
- 5- Soit l'affectation suivante :
Nom ← "Beldi"
a- Quelles sont les valeurs de Nom[1] et de Nom[5] ?
b- Après l'affectation Nom[2] ← "a" , Quel est le contenu de la variable Nom.

Réponses :

- 1- Tableau de déclaration des objets

Objet	Type/Nature	Rôle
Prenom	Chaîne [10]	Prénom d'un individu sur 10 caractères maximum
Nom	Chaîne	Chaîne pouvant contenir jusqu'à 255 caractères.
Adresse	Chaîne [25]	Chaîne pouvant contenir jusqu'à 25 caractères.

- 2- Dans le langage PASCAL les déclarations correspondantes seront :

VAR

```
Prenom : String [10] ;
Nom : String ;
adresse : String [25] ;
```

Remarque :

À la variable Prenom, on peut affecter une chaîne de caractères ayant au maximum dix caractères. Par contre, il n'y a pas de limitation visible pour la longueur de la variable Nom (Ce nombre maximal est de 255 caractères pour certains compilateurs).

- 3- Prenom ← "" {vide}
 Nom ← "Aloui"
 Adresse ← " " {un espace}
- 4- On pourra accéder en lecture et en écriture au ième caractère d'une chaîne CH en utilisant la notation CH[i] où $1 \leq i \leq \text{Long}(\text{CH})$ où Long(CH) désigne la longueur de la chaîne CH.
- 5- a- Nom [1] donne "B"
 Nom [5] donne "i"
 b- Nom devient "Baldi"

Les fonctions et les procédures prédéfinies

Les langages de programmation offrent un jeu de fonctions et de procédure prédéfinies. Nous étudions dans la suite les fonctions et les procédures les plus usuelles.

Activité 13

- 1- Soient les exemples de fonctions et procédures prédéfinies suivantes :

concat ("Bon","jour")	Long ("Lycée")
Sous_chaîne ("Baccalauréat",6,7)	Pos ("Bac","Baccalauréat")
Convch (2008,ch)	Valeur ("16.5",d,erreur)
Efface (v, 7,3) où v contient "Disquette"	
Insère ("tte",v,7) où v contient "Disque"	

Elaborer un tableau de 4 colonnes donnant le nom algorithmique de chacune des procédures ou fonctions précédentes, le code en pascal, le rôle de chacune des procédures ou fonctions et les résultats générés par les exemples définis ci-dessus.

Algorithmique	Code en Pascal	Rôle	Exemples
Concat(ch1,ch2, ...,chN)	CONCAT (ch1, ch2, ... chN)	Concat retourne la concaténation des chaînes ch1, ch2, ..., chN	concat ("Bon","jour") retourne la chaîne "Bonjour"
Long (ch)	LENGTH (ch)	Long retourne un entier représentant la longueur en caractères de la chaîne ch. Ce nombre se trouve aussi dans le caractère ch[0].	Long ("Lycée") retourne l'entier 5
Sous_chaîne (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.	Sous_chaîne("Baccalauréat",6,7) retourne la chaîne "lauréat"

Pos (ch1, ch2)	POS (ch1, ch2)	Pos retourne la première position de la chaîne ch1 dans la chaîne ch2	Pos("Bac","Baccalauréat") retourne l'entier 1
Efface (ch, p, n)	DELETE (ch, p, n)	Efface est une procédure qui enlève n caractères de ch à partir de la position p .	Efface(v, 7,3) modifie v qui contiendra "Disque"
Insère(ch1,ch2, p)	INSERT(ch1, ch2, p)	Procédure qui insère la chaîne ch1 dans la chaîne ch2 à partir de la position p . Le caractère numéro p et les suivants sont décalés vers la droite.	Insère ("tte",v,7) modifie v qui contiendra "Disquette"
Convch (d, ch1)	STR (d, ch1)	Procédure qui convertit un nombre décimal d en chaîne de caractères et l'affecte à la variable ch1 .	Convch(2008, ch) modifie ch qui contiendra "2008"
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. Le paramètre erreur est une variable de type entier qui contiendra 0 si la conversion s'est bien déroulée, sinon elle contiendra la position du caractère qui a déclenché l'erreur.	Valeur("16.5",d,erreur) modifie d qui contiendra le réel 16.5. erreur contiendra 0.

Retenons

- Les constantes contiennent des valeurs inchangées tout le long de l'exécution de l'algorithme. Les variables peuvent changer de contenu tout le long de l'algorithme.
- Un type définit un ensemble de valeurs et un ensemble d'opérations applicables sur ces valeurs.
- Les types standard sont :
 - les entiers,
 - les réels,
 - les booléens,
 - les caractères,
 - et les chaînes de caractères

Exercices

Exercice 1

Quelle est la différence entre une constante et une variable?

Exercice 2

Évaluer le contenu des variables m , n , p suite à l'exécution de chacune des séquences suivantes :

Séquence 1	Séquence 2
1) $m \leftarrow 30$	1) $m \leftarrow 2$
2) $n \leftarrow m + 20$	2) $m \leftarrow m * m$
3) $p \leftarrow n + m$	3) $m \leftarrow m * m$
4) $m \leftarrow n - p$	4) $m \leftarrow m * m$

Exercice 3

1- Évaluer le contenu des variables a , b , et c suite à l'exécution de chacune des séquences suivantes :

Séquence 1	Séquence 2	Séquence 3
1) $a \leftarrow 3$	1) $a \leftarrow 3$	1) $a \leftarrow 7$
2) $b \leftarrow 5$	2) $b \leftarrow 5$	2) $b \leftarrow 3$
3) $a \leftarrow b$	3) $c \leftarrow b$	3) $b \leftarrow a + b$
4) $b \leftarrow a$	4) $b \leftarrow a$	4) $a \leftarrow b - a$
	5) $a \leftarrow c$	5) $b \leftarrow b - a$

2- Quel est le rôle de la variable c dans la séquence 2 ?

Exercice 4

On se propose de calculer la surface d'un rectangle. Etablir un tableau de déclaration des objets nécessaires.

Exercice 5

Dire si l'utilisation de ces identificateurs est permise dans le langage PASCAL ou non. Justifier votre réponse.

- Code Produit
- Code+Produit
- Code_Produit
- 3etapes
- capacité

Exercice 6

Le programme Pascal suivant comporte de nombreuses erreurs, trouver-les.

```

PROGRAM deserreurs ;
  CONST
    I=10 ; J=40 ; K=5 ;
  TYPE
    integer = -32000..+32000
    voyelles1=('A','E','I','O','U');
    voyelles2=(A,E,I,O,U);
    Abscisses=0..0.001;
    Indice= -10..+10;
  VAR
    S:Indice ; V: voyelle2 ; R: REAL;
  BEGIN
    R:=35;
    V:=R+1;
    S:=2*J+K;
  END

```

Exercice 7

Dire si l'on pourra utiliser le type entier pour représenter les quantités suivantes :

- a- La note obtenue dans un examen
- b- Le nombre de matières étudiées pendant un trimestre
- c- Le coefficient d'une matière
- d- La moyenne générale du trimestre

Exercice 8

Ecrire les instructions PASCAL permettant de réaliser les objectifs suivants :

- 1- Obtenir la longueur de la chaîne "Informatique"
- 2- Récupérer la sous-chaîne de 5 caractères à partir de la position 8 de la chaîne "Informatique"
- 3- Effacer les 5 premiers caractères de la chaîne "Informatique"

Exercice 9

a est un entier, x est un réel et ch est un caractère.

Dire pourquoi les instructions suivantes sont erronées, et les corriger dans la mesure du possible.

- a:=3.4
- x:=5,16
- ch:=x
- a:=maxent+5

Exercice 10

Soit la partie déclarative du programme Pascal suivant :

```
PROGRAM Dates;
  CONST
    Date1="03/07/2006";
  VAR
    Date2: STRING[10];
    Jour: STRING[2];
    Mois: STRING[2];
    Annee: STRING[4];
    .....
```

Compléter le corps du programme par les instructions (bloc BEGIN...END) permettant de réaliser les traitements suivants :

- Mettre Date1 dans Date2 sous la forme jjmmaaaa (indication: Date2 va contenir 03072006)
- Mettre le jour dans la variable Jour, le mois dans la variable Mois et l'année dans la variables Annee.

Exercice 11

Soit la partie déclarative du programme Pascal suivant :

```
PROGRAM mots ;
  CONST
    m1: "abcd";
    m2: "ordinateur";
    m3: "scalaire";
  VAR
    symetrie: STRING[8];
    compose: STRING ;
    .....
```

Compléter la partie du programme Pascal précédente en ajoutant des variables (si c'est nécessaire) et le bloc BEGIN ... END pour :

- Avoir dans la variable « symetrie » une chaîne de caractères symétrique obtenue à partir de la variable « m1 ».
- Avoir dans la variable « compose » une chaîne de caractères obtenue en combinant la première moitié de « m2 » et la deuxième moitié de « m3 » en utilisant les fonctions : concat() et copy().

Exercice 12

Ecrire un programme Pascal qui permet de transformer la chaîne de caractères "informatique" en majuscule sans utiliser la fonction upcase(c), et la mettre dans une variable nommée maj.

Indication : le code ASCII de 'a' est 97 et le code ASCII de 'A' est 65.

Leçon 2

Les expressions

Objectifs spécifiques :

- Reconnaître les expressions
- Evaluer des expressions en tenant compte de la priorité des opérateurs
- Utiliser des fonctions prédéfinies dans des expressions

Plan de la leçon :

I. Les opérandes

II. Les opérateurs

- II.1. Les opérateurs arithmétiques
- II.2. Les opérateurs logiques
- II.3. Les opérateurs relationnels

Retenons

Exercices

Leçon 2

Les expressions

J'appelle claire (la connaissance) qui est présente et manifeste à un esprit attentif : de même que nous disons voir clairement les objets lorsque étant présents ils agissent assez fort, et que nos yeux sont disposés à les regarder.

DESCARTES

Une expression simple est constituée d'opérandes reliés par des opérateurs. Nous distinguons deux types d'expressions :

- les expressions arithmétiques donnant une valeur numérique.
- les expressions logiques donnant une valeur booléenne.

Activité 1

Donner le type et la valeur de chacune des expressions suivantes.

1- $(6 < 2)$

2- $(7+3) \text{ DIV } 2$

Réponses :

1- C'est une expression logique ; le résultat est booléen ; sa valeur est FAUX.

2- C'est une expression arithmétique ; le résultat est entier ; sa valeur est 5.

I. Les opérandes

Activité 2

1- Soient les expressions suivantes :

a- $5 + 3.141$

b- $7 \text{ MOD } 3$

Indiquer pour chacune des expressions ci-dessus le premier et le deuxième opérande et l'opérateur.

2- Quels sont les types d'opérandes ?

3- Donner les valeurs résultats de chacune de ces expressions :

a- $3.14 * D1$

b- $(3.14 * D1) + (3.14 * D2)$

c- $3.14 * \text{CARRE}(R1)$

d- $100/25$

pour les valeurs $D1=4$, $D2=3$ et $R1=2$

Réponses :

1- a- 5 est le premier opérande ; + est l'opérateur et 3.141 constitue le deuxième opérande.

b- 7 est le premier opérande ; MOD est l'opérateur et 3 constitue le deuxième opérande.

2- Les opérandes peuvent être des constantes, des variables, des valeurs ou des résultats envoyés par des fonctions. Par ailleurs, un opérande peut être une expression.

3- a- 12.56

b- 21.98

c- 12.56

d- 4.0

II. Les opérateurs

II.1 Les opérateurs arithmétiques

Les opérateurs unaires

Un opérateur est dit unaire s'il est appliqué à un seul opérande. On dit aussi qu'il est monadique.

L'opérateur unaire usuel

Exemple : $-(77)$ - est l'opérateur et **77** est l'opérande.

Les opérateurs binaires

Un opérateur est dit binaire s'il est appliqué à deux opérandes. On dit aussi qu'il est dyadique.

Opérateur	Type opérande	Type résultat
-	Entier ou Réel	Entier ou Réel

Activité 3

- 1- Remplir un tableau permettant de montrer le type du résultat pour les différents types d'opérandes admis en ce qui concerne les opérateurs binaires multiplicatifs.
- 2- Remplir un tableau permettant de montrer le type du résultat pour les différents types d'opérandes admis en ce qui concerne les opérateurs binaires additifs.

Réponses :

Opérateur	type opérande 1	type opérande 2	type résultat
*	Entier	Entier	Entier
	Réel	Réel	Réel
	Réel	Entier	Réel
	Entier	Réel	Réel
/	Entier	Entier	Réel
	Entier	Réel	Réel
	Réel	Entier	Réel
	Réel	Réel	Réel
DIV	Entier	Entier	Entier
MOD	Entier	Entier	Entier

Opérateur	type opérande 1	type opérande 2	type résultat
+,-	Entier	Entier	Entier
	Entier	Réel	Réel
	Réel	Entier	Réel
	Réel	Réel	Réel

II.2 Les opérateurs logiques

Activité 4

- 1- a- Quel est l'opérateur unaire usuel, le type de l'opérande et le type du résultat?
b- Evaluer l'expression suivante : NON (44<66)
- 2- Remplir un tableau permettant de montrer le type du résultat pour les types d'opérandes admis en ce qui concerne les opérateurs binaires logiques.

Réponses :

1- a-

Opérateur	Type opérande	Type résultat
NON	Booléen	Booléen

b- NON est l'opérateur et (44<66) est un opérande. Le résultat de cette expression est FAUX.

2-

Opérateurs	type opérande 1	type opérande 2	type résultat
ET	Booléen	Booléen	Booléen
OU	Booléen	Booléen	Booléen
Ouex	Booléen	Booléen	Booléen

NB : Souvent les opérateurs logiques sont notés multiplicativement pour le ET et additivement pour le OU.

II.3 Les opérateurs relationnels

Le tableau suivant illustre les opérateurs relationnels usuels :

Opérateurs	Code en Pascal	Type opérande 1	Type opérande 2	Type résultat
<, >, =, <>, <=, ≥	<, >, =, <>, <=, >=	Tout type ordonné	Tout type ordonné	Logique

Remarque :

Tous les types que nous avons vus sont des types ordonnés. Toute comparaison entre deux éléments de même type ou de types compatibles est possible.

Activité 5

- 1- Quel est le résultat d'une comparaison de deux éléments de même type ou de types compatibles ?
- 2- Pour x=32 et y=36 , quelle est la valeur de x<y ?

Réponses :

- 1- Le résultat de la comparaison est booléen (VRAI ou FAUX).
- 2- $x < y$ vaut VRAI

III. Évaluation d'une expression

Lors de l'évaluation d'une expression, on tient compte de la priorité entre les opérateurs.

Activité 6

- 1- Quel est l'ordre de priorité dans le calcul des expressions ?
- 2- Donner les étapes de calcul de l'expression $31+7*10$.
- 3- Evaluer l'expression $55 + 6 - 10$
- 4- Dans le cas où on veut imposer un autre ordre, que doit-on faire ?
- 5- Evaluer les expressions suivantes :
 - a- $(44 * x + y)$ avec $x = 2$ et $y = 4$
 - b- $(y + 44 * x)$ avec $x = 2$ et $y = 4$
 - c- $((y + 44) * x)$ avec $x = 2$ et $y = 4$

Réponses :

- 1- Cet ordre est le suivant :
 - 1) Les parenthèses.
 - 2) Les opérateurs unaires.
 - 3) Les opérateurs multiplicatifs.
 - 4) Les opérateurs additifs.
 - 5) Les opérateurs relationnels.
- 2- 1^{ère} opération $7 * 10 = 70$ 2^{ème} opération $31 + 70 = 101$
- 3- Le calcul de $55 + 6 - 10$ commence par $55 + 6$ donc 61 puis $61 - 10$ pour avoir 51.
En effet, pour les opérateurs de même priorité, on commence par celui qui est le plus à gauche.
- 4- Dans ce cas, on doit utiliser des parenthèses.
 $31 + 7 * 10$ vaut 101 mais si on écrit $(31 + 7) * 10$ le résultat est 380.
- 5- a- 92 b- 92 c- 96

Retenons

Une expression est composée d'opérateurs et d'opérandes et son évaluation produit une valeur.

- On distingue les expressions arithmétiques, et les expressions logiques.
- L'évaluation d'une expression se fait toujours selon l'ordre de priorité des opérateurs.

Exercices

Exercice 1

1- Donner les expressions arithmétiques correspondantes aux expressions suivantes écrites en Pascal :

`sqrt(2 * a + 3 / b - 2) / 3 + x`

`4 * x / c * 6 - x`

`10 * x / 2 + 4`

2- Réciproquement, écrire en Pascal les expressions arithmétiques suivantes :

$$\frac{4x^2+2x-5}{\sqrt{x+\frac{5y}{2}}}$$

$$\frac{y+2}{10x} + 1$$

$$\sqrt{\frac{x^2-y}{y-\frac{x}{2}}}$$

Exercice 2

Evaluer les expressions logiques en A, B, C, et D pour chacune des combinaisons (p,q,r,s) suivantes :

1) Pour (p,q,r,s) = (-3, 5, 4, 9)

2) Pour (p,q,r,s) = (3, 7, 4, 9)

3) Pour (p,q,r,s) = (5, 13, 7, 3)

A. (p < q) OU (r > s)

B. (p < q) ET NON (r > s)

C. (p > q) OU (r ≠ p)

D. (p + q < r) ET (p + q > r)

Exercice 3

Cette écriture permettant de vérifier si le caractère C est une voyelle est erronée . Pourquoi ? Qu'aurait-il fallu écrire ?

`C='A' OR C='E' OR C='I' OR C='O' OR C='U'`

Exercice 4

- Rappeler l'ordre de priorités dans lequel une expression doit être évaluée.
- Comment évaluer deux opérateurs ayant la même priorité ?
- Compléter le tableau suivant par le type du résultat :

Opérateur	Type opérande 1	Type opérande 2	Type du résultat
DIV	entier	entier
/	entier	entier
+	réel	entier
-	entier	entier

Leçon 3

Le type scalaire énuméré et le type intervalle

Objectifs spécifiques :

- Comprendre l'utilisation du type énuméré.
- Comprendre l'utilisation du type intervalle.
- Manipuler des variables faisant appel aux types énuméré et intervalle.

Plan de la leçon :

I. Le type scalaire énuméré

II. Le type intervalle

Retenons

Exercices

Leçon 3

Le type scalaire énuméré et le type intervalle

La conscience des représentations qui suffit pour différencier un objet d'un autre. C'est la clarté. Mais celle qui rend claire la composition des représentations, c'est la distinction.

KANT

Outre les types standard présentés dans la première leçon, nous pouvons définir de nouveaux types appelés souvent types utilisateur.

I. Le type scalaire énuméré

Dans certaines situations où l'on aimerait contraindre une variable à décrire un jeu de valeurs bien déterminé, on utilisera un type scalaire énuméré afin d'énumérer ces valeurs.

Un type scalaire définit, en général, un domaine de valeurs comme c'est le cas du type entier. Par ailleurs, le type scalaire par énumération définit un ensemble ordonné et fini de valeurs désignées par des identificateurs.

Définition

Le type scalaire par énumération définit un ensemble ordonné et fini de valeurs désignées par des identificateurs

Activité 1

Nous voulons définir un type que nous appelons ANNEE_SCOLAIRE contenant les dix mois de l'année scolaire et manipuler les valeurs de ce type.

- 1- Donner le tableau de déclaration des nouveaux types illustrant le type ANNEE_SCOLAIRE.
- 2- L'affectation suivante est-elle correcte sachant que mois est une variable de type ANNEE_SCOLAIRE ?
`mois ← octobre`
- 3- Quel est l'ordre de ces constantes ?
- 4- Quels sont les opérateurs applicables à ces valeurs ?
- 5- Evaluer les expressions suivantes :
 - a- SUCC (septembre)
 - b- PRED (juin)
- 6- La déclaration du type énuméré IMPAIR = (1, 3, 5, 7) est-elle possible ?
- 7- Déclarer en Pascal
 - a- un type scalaire énuméré contenant les mois à 30 jours.
 - b- une variable intitulée mois_court du type mois_a_trente.

Réponses :**1- Tableau de déclaration des nouveaux types**

Types
ANNEE_SCOLAIRE=(septembre, octobre, novembre, decembre, janvier, fevrier, mars, avril, mai, juin)

septembre, octobre, novembre, decembre, janvier, fevrier, mars, avril, mai, juin sont les éléments du type ANNEE_SCOLAIRE.

2- Une variable mois de type ANNEE_SCOLAIRE peut prendre comme valeur : septembre, octobre, novembre, decembre, janvier, fevrier, mars, avril, mai, juin. L'affectation mois ← octobre est correcte.

3- L'ordre sur ces valeurs est le suivant : septembre < octobre < ... < juin. Par ailleurs, on peut appliquer la fonction Ord sur ces valeurs pour déterminer leurs numéros d'ordre, ainsi Ord (septembre) vaut 0 et Ord (octobre) vaut 1 et ainsi de suite.

4- Les opérateurs applicables à ces valeurs sont :
 • Les opérateurs de relation
 • Les opérateurs **PRED** et **SUCC**. **PRED** représente le prédécesseur (le précédent) et **SUCC** représente le successeur (le suivant).

5- a- octobre

b- mai

Remarque : **PRED** (septembre) n'existe pas et **SUCC** (juin) n'existe pas non plus et peuvent provoquer des erreurs.

6- Cette déclaration est interdite car 1, 3, 5 et 7 sont des valeurs qui appartiennent au type prédéfini ENTIER.

7- a- **TYPE** mois_a_trente = (avril, juin, septembre, novembre) ;
 b- **VAR** mois_court : mois_a_trente ;

Dans le langage Pascal, la déclaration d'un type scalaire énuméré et celle d'une variable de ce type se font comme suit :

```
TYPE <nom_type>=(constante_1, constante_2, ... ,constante_n);
VAR <nom_variable> : nom_type ;
```

II. Le type intervalle

Le type intervalle possède les propriétés d'un type scalaire discret ordonné (entier, caractère et scalaire énuméré).

La définition d'un intervalle est décrite par la donnée de deux constantes représentant respectivement la "Borne Inférieure" et la "Borne Supérieure" appartenant à un type scalaire discret ordonné et telle que Borne Inférieure < Borne Supérieure.

Activité 2

- 1- Elaborer un tableau de déclaration des nouveaux types illustrant un type intervalle intitulé mois de 1 à 12 et un type intervalle concernant le premier trimestre de l'année scolaire.
- 2- a- Quels sont les bornes inférieure et supérieure du type mois ?
 b- Quels sont les valeurs que peut prendre une variable du type mois ?
 c- Quels sont les bornes inférieure et supérieure du type premier_trimestre ?
 d- Quels sont les valeurs que peut prendre une variable du type premier_trimestre?
- 3- Déclarer en Pascal un type intervalle mois, un type intervalle jours et deux variables mois_actuel et j de types respectifs mois et jours.

Réponses :

1- Tableau de déclaration des nouveaux types

Types
Mois = 1 .. 12
ANNEE_SCOLAIRE = (septembre, octobre, novembre, decembre, janvier, fevrier, mars, avril, mai, juin)
Premier_trimestre = septembre .. decembre

- 2- a- Les bornes de Mois sont 1 et 12 prises comme valeurs entières.
 b- Une variable de type Mois peut prendre ses valeurs entre 1 et 12.
 c- Les bornes de premier_trimestre sont septembre et decembre.
 d- Une variable de type premier_trimestre peut prendre comme valeur : septembre, octobre, novembre et decembre.

- 3- **TYPE** mois = 1 .. 12;
 jours = 1 .. 31 ;
 VAR mois_actuel : mois ;
 j : jours ;

En effet, dans le langage Pascal, la déclaration d'un type intervalle et celle d'une variable de ce type se font comme suit :

```
TYPE <nom_type> = borne_inf .. borne_sup ;
VAR <nom_variable> : nom_type ;
```

Remarques :

Une variable d'un type intervalle possède toutes les propriétés du type de base dont l'intervalle est issu. Toutefois, sa valeur doit être comprise au sens large entre les bornes de l'intervalle.

L'intérêt de ce type réside dans le fait qu'il permet une meilleure lisibilité de l'algorithme et du programme.

Exemple : mois : 1 .. 12 est beaucoup plus précis que mois : entier

Retenons

Le type scalaire par énumération définit un ensemble ordonné et fini de valeurs désignées par des identificateurs définis par l'utilisateur.

Les opérateurs applicables à ces valeurs sont :

- Les opérateurs de relation
- Les opérateurs PRED et SUCC.

Le type intervalle possède les propriétés d'un type scalaire discret ordonné (entier, caractère et scalaire énuméré).

La définition d'un intervalle est décrite par la donnée de deux constantes représentant respectivement la "Borne Inférieure" et la "Borne Supérieure" appartenant à un type scalaire discret ordonné et telle que Borne Inférieure < Borne Supérieure.

Exercices

Exercice 1

Soit le programme Pascal suivant :

```

PROGRAM erreurs
  TYPE
    eleves   : (Ali,Safa,Sami,Wissem,Kamel) ;
    moyenne = 0..20
  VAR
    e1: eleves;
    e2: eleves;
    n1: moyenne
    n2 : moyenne ;
    reussir : BOOLEAN ;

  BEGIN
    e1 := "Safa";
    n1 = ORD(Kamel)+2 * SUCC(Ali)  ;
    e2 := Sami  ;
    n2 := -15;
    russir := ( n2 > 10 )  ;
  END.

```

- 1) Corriger les erreurs du programme Pascal ci-dessus.
- 2) Evaluer le contenu des variables utilisées dans le Programme.

Exercice 2

Est-ce que la déclaration de l'énumération suivante est correcte? Justifier votre réponse.

```
Pair = (0,2,4) ;
```

Exercice 3

En utilisant le type intervalle, déclarer en algorithmique et en Pascal les variables suivantes :

- jour
- mois

Exercice 4

En utilisant le type scalaire énuméré, déclarer en algorithmique puis en Pascal les types suivants :

- couleur_de_base
- jour_de_la_semaine

Leçon 4

Les tableaux à une dimension

Objectifs spécifiques

- Comprendre l'utilisation du type tableau.
- Manipuler des tableaux.

Plan de la leçon

I. Déclaration d'un tableau

II. Le type tableau

Retenons

Exercices

Leçon 4

Les tableaux à une dimension

Si, dans la représentation, on laisse de côté les déterminations d'un objet, c'est ce q'on appelle abstraire. Il ne reste alors qu'un objet moins déterminé. C'est à dire un objet abstrait. Mais, si dans la représentation je ne considère q'une détermination singulière de cet ordre, c'est là aussi une représentation abstraite.

HEGEL

Quand on veut regrouper un certain nombre de variables de même type sous un même nom, on utilise la notion de tableau. Un tableau peut être considéré comme une suite de variables, de même nom, repérées par des indices.

Définition :

Un tableau est une structure de données homogènes regroupant un ensemble d'éléments de même type.

I. Déclaration d'un tableau

Dans le cas général, pour déclarer un tableau, on utilisera la forme suivante :

Au niveau de l'analyse et de l'algorithme

Tableau de déclaration des objets

Objet	Type nature	Rôle
Ident_tableau	Tableau de Taille et de Type_élément	

En Pascal

VAR

```
ident_tableau: ARRAY[Borne_inf..Borne_sup] OF Type_élément;
```

Où :

Ident_tableau : Identificateur du nouveau tableau que nous voulons définir.

Borne_Inf .. Borne_Sup : intervalle correspondant à l'ensemble des valeurs des indices du tableau.

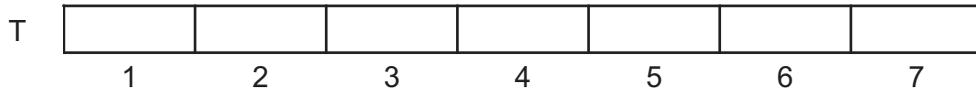
Borne_Inf : Borne inférieure de l'intervalle des indices.

Borne_Sup : borne supérieure de l'intervalle des indices.

Type_élément : type des éléments du tableau. Il peut être l'un des types vu précédemment.

Exemple :

T : **ARRAY** [1..7] **OF REAL** ; {déclaration d'un tableau de 7 éléments réels }

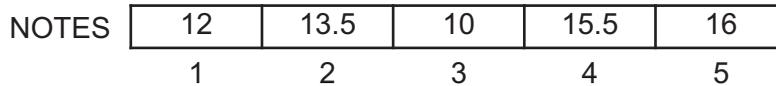


Suite à cette déclaration, nous réservons un espace mémoire au niveau de la RAM pouvant contenir sept réels.

Un tableau est caractérisé par ses dimensions. Nous nous limiterons à l'étude **des tableaux unidimensionnels** appelés aussi vecteurs.

Activité 1

- 1- Donner le tableau de déclaration des objets pour un vecteur de 5 éléments de type réel pouvant contenir les notes des élèves.
- 2- Déclarer le tableau NOTES en Pascal.
- 3- Affecter à chaque élément la note correspondante suivant le tableau suivant :



Réponses :

- 1- La déclaration du tableau se fera comme suit :

Tableau de déclaration des objets

Objet	Type nature	Rôle
NOTES	Tableau de 5 réels	Tableau servant à contenir les notes de 5 élèves

- 2- Dans le langage Pascal, on aurait effectué la déclaration suivante :

VAR

NOTES : **ARRAY**[1..5] **OF REAL**;

- 3- NOTES [1] est le 1^{er} élément du tableau NOTES.
 NOTES [2] est le 2^{ème} élément du tableau NOTES.
 NOTES [3] est le 3^{ème} élément du tableau NOTES.
 NOTES [4] est le 4^{ème} élément du tableau NOTES.
 NOTES [5] est le 5^{ème} élément du tableau NOTES.

NOTES [1] ← 12
 NOTES [2] ← 13.5
 NOTES [3] ← 10
 NOTES [4] ← 15.5
 NOTES [5] ← 16

Remarques :

- 1) Pour accéder au ième élément du tableau, il suffit de donner l'identificateur du tableau et l'indice i indiquant le rang de l'élément. Cet indice doit être dans l'intervalle Borne_inf.. Borne_sup.
- 2) Nous pouvons lire et écrire un élément du tableau (voir chapitre 2). Nous pouvons modifier un élément par une instruction d'affectation.
Exemple : NOTES [4] ← 17.50
- 3) Les opérations possibles sur un élément du tableau sont les mêmes que celles définies sur une variable de même type.

II. Le type tableau

Il est possible de déclarer le type d'un tableau.

Exemple**Tableau de déclaration des nouveaux types**

Types
Elevés = tableau de 30 chaînes de caractères
Moyennes = tableau de 30 réels
Comptes = tableau de 26 entiers

Tableau de déclaration des objets

Objet	Type / Nature	Rôle
T_ELEVES	Elevés	Tableau des 30 noms d'élèves
T_MOY	Moyennes	Tableau des 30 moyennes d'élèves
T_Compte	Comptes	Tableau comptant le nombre de chaque lettre de 'A' à 'Z' du texte

Activité 2

Soit la séquence suivante :

```

A [ 1 ] ← 10
A [ 2 ] ← 7
A [ 3 ] ← A [ 1 ] Div A [ 2 ]
A [ 4 ] ← A [ 3 ] A [ 2 ]
A [ 5 ] ← A [ 1 ] + A [ 3 ] * A [ 4 ]

```

- 1- Déclarer le tableau A.
- 2- Quel est le contenu de chaque élément du tableau A ?

Réponses

1- Déclaration

Tableau de déclaration de nouveaux types

Types
vecteur = tableau de 5 entiers

Tableau de déclaration des objets

Objet	Type / Nature	Rôle
A	vecteur	Tableau de 5 éléments

2- Le contenu de chaque élément :

A	10	7	1	7	17
	1	2	3	4	5

Retenons

- Les structures de données vues dans cette leçon sont les tableaux (suite de variables de même type repérées par des indices).
- Avant d'utiliser un tableau, il faut le déclarer.
- Il faut bien faire la différence entre l'indice d'un élément et le type des éléments.

Exercices

Exercice 1

Elaborer un tableau de déclaration des objets relatifs aux :

- noms des employés
- leurs nombres de jours travaillés
- leurs salaires

d'une entreprise comptant 50 employés.

Exercice 2

- 1- Déclarer le tableau **JOURS** qui contient les sept jours de la semaine.
- 2- De même, déclarer le tableau **MOIS** permettant de regrouper les douze mois de l'année.

Exercice 3

Soit le tableau suivant :

T	14	10	19	84	92
---	----	----	----	----	----

- 1- Déclarer le tableau T en algorithmique et en Pascal.
- 2- Donner les affectations permettant de remplir T.
- 3- Inverser les éléments du tableau T sans utiliser un autre tableau.

Exercice 4

Soit le tableau de SALAIRES exprimés en Dinars.

SALAIRES	325.560	650.800	720.252	529.100	390.440
	1	2	3	4	5

- 1- Donner les cinq affectations permettant de remplir le tableau SALAIRES par les données précédentes.
- 2- Ranger, dans un deuxième tableau SALAIRES_MI, les équivalences des éléments de SALAIRES en millimes.

Exercice 5

- 1- Soit V et W deux tableaux d'entiers, de types respectifs T1 et T2 et de tailles respectives 4 et 3.
- 2- Soit la séquence d'affectations suivantes :
 - V[1] ← 300
 - V[2] ← 50
 - W[1] ← V[1] + V[2]* 2

```

V[3] ← W[1] DIV 3
W[2] ← W[1] MOD V[1]
W[3] ← V[2] * 2 + 2
V[4] ← (V[2] DIV 4) MOD 2
V[8] ← V[4]

```

Questions

- 1- Déclarer les deux tableaux V et W.
- 2- Trouver les erreurs d'affectation dans la séquence précédente.
- 3- Quel est le contenu de chaque élément des deux tableaux V et W ?

Exercice 6

Trouver les erreurs du programme Pascal suivant :

```

PROGRAM pleinerreurs ;
  CONST
    N=5 ;
  VAR
    I, J, N, M : INTEGER ;
    C: CHAR;
    A: ARRAY[1..N] of CHAR;
    B: ARRAY[1..M] of CHAR;
  BEGIN
    I:=0;
    C:='0';
    M:=4;
    A[1]:=C;
    A[2]:= SUCC(C);
    B[1]:=A[1];
  END.

```

Exercice 7

Soit le tableau **T** de mots dans lequel les mots sont classés par longueur (d'abord tous les mots d'une lettre, puis de deux lettres,..., puis de 8 lettres (taille maximale du mot)).

Un deuxième tableau, **CLE**, contient pour un indice *i* l'indice dans **T** du premier mot de longueur *i*. Que contient **CLE[j]** s'il n'y a pas dans **T** de mot de longueur *j* ? Donner un exemple contenant les déclarations et les affectations nécessaires.