

# CHAPITRE

## Chapitre 2

### Les actions élémentaires simples

#### Objectifs :

- Utiliser les structures simples pour résoudre des problèmes.
- Écrire des programmes en Pascal utilisant les structures simples.

#### Plan du chapitre :

Leçon 1 :  
L'affectation

Leçon 2 :  
Les opérations d'entrée / sortie



## Leçon 1

# L'affectation

### Objectifs spécifiques :

- Utiliser les structures simples pour résoudre des problèmes.
- Présenter les solutions sous forme d'un algorithme puis d'un programme.

### Plan de la leçon :

#### I. Introduction

#### II. Définition

#### III. Vocabulaire et syntaxe

#### Retenons

#### Exercices

## Leçon 1

## L'affectation

*J'entends par attribut ce que l'entendement perçoit d'une substance comme constituant son essence.*

SPINOZA

## I. Introduction

Dans cette leçon, nous allons revenir plus en détail sur l'action élémentaire intitulée "**affectation**". Comme nous l'avons vu au chapitre précédent, l'affectation est l'action ou l'instruction élémentaire de base permettant de modifier la valeur d'une variable. Quand nous savons que la notion de variable est fondamentale dans l'exercice de la programmation, nous comprenons l'importance que peut prendre l'action d'affectation.

## Activité 1

Soit la séquence d'instructions suivante :

- 1)  $X \leftarrow 2$
- 2)  $Y \leftarrow X * X$
- 3)  $Y \leftarrow Y * Y$
- 4)  $Y \leftarrow Y * X$
- 5)  $Y \leftarrow Y * Y$

- 1- Dresser un tableau (traces de la séquence algorithmique) pour déterminer les valeurs des variables  $X$  et  $Y$  après l'exécution des instructions précédentes.
- 2- Quel est le rôle de cette séquence d'instructions.

## Réponse :

- 1- Trace de l'exécution des instructions précédentes

N° instruction	X	Y
1	2	
2	2	4
3	2	16
4	2	32
5	2	1024

- 2- Cette séquence d'instructions permet de calculer à partir d'une valeur  $X$ , la valeur  $Y$  correspondante. Après la dernière instruction,  $Y$  vaut  $X^{10}$ .

## II. Définition

L'opération d'affectation consiste à attribuer une valeur à une variable. L'instruction d'affectation se note avec le symbole " $\leftarrow$ " en algorithmique et "==" en Pascal.

L'expression à droite du symbole d'affectation sera évaluée puis le résultat sera affectée à la variable située à gauche du symbole " $\leftarrow$ ".

### Activité 2

Sachant que m, n, p et r sont quatre variables de type entier ou de type entier long. Soit la séquence ci-dessous :

- 1)  $m \leftarrow 159383552$
- 2)  $n \leftarrow m \text{ DIV } 8$
- 3)  $p \leftarrow n \text{ DIV } 1024$
- 4)  $r \leftarrow p \text{ DIV } (1024)$

- 1- Déterminer la valeur de chacune des variables n, p et r après l'exécution de la séquence ci-dessus.
- 2- Que fait cette séquence d'instructions?
- 3- Traduire cet algorithme en Pascal.

**Réponse :**

- 1- - La valeur de n vaut 19922944.  
- La valeur de p vaut 19456.  
- La valeur de r vaut 19.
- 2- Cette séquence convertit une capacité mémoire donnée en bits en son équivalent en octets, kilo octets et méga octets.

### Traduction en Pascal

```
PROGRAM CONVERSION;
USES WINCRT;
VAR
    m,n,p,r : LONGINT;

BEGIN
    m := 159383552;
    n := m DIV 8;
    p := n DIV 1024;
    r := p DIV 1024;
END.
```

Vous allez remarquer qu'en exécutant ce programme, vous n'allez rien voir à l'écran. Dans la leçon suivante, nous allons étudier comment afficher des messages et des résultats à l'écran.

### III. Vocabulaire et syntaxe

	Au niveau de l'analyse et de l'algorithme	Au niveau du Pascal
Forme générale	Variable $\leftarrow$ expression	Variable := expression;

**N.B.** Une expression peut être une valeur.

**Exemples :**

Analyse et algorithme	Pascal	Commentaire
$Y \leftarrow 7$	$Y := 7;$	La variable Y reçoit la valeur 7.
$Ch \leftarrow \text{"cible"}$	$Ch := \text{'cible'};$	La chaîne Ch reçoit la chaîne cible.
$X \leftarrow Y$	$X := Y;$	La valeur de la variable X devient égale à la valeur de la variable Y.
$L \leftarrow \text{long}(Ch) \text{ DIV } 2$	$L \leftarrow \text{length}(Ch) \text{ DIV } 2;$	La valeur de la variable L devient égale à la longueur de la chaîne Ch divisée par deux.
$Y \leftarrow Y + 1$	$Y := Y + 1;$	La valeur de la variable Y devient égale à sa valeur actuelle incrémentée de 1.

**Remarques :**

- L'expression est évaluée avant d'être affectée à la variable.
- La valeur de l'évaluation de l'expression doit être compatible avec le type de la variable ; sinon l'exécution de cette affectation provoquera une erreur.
- Il est possible d'affecter la valeur d'une variable à une autre variable.
- Le nom d'une variable dans une expression signifie sa valeur actuelle.
- L'instruction d'affectation ne modifie que ce qui est situé à gauche du symbole d'affectation " $\leftarrow$ ".

#### Activité 3

Effectuer une analyse, un algorithme et la traduction en Pascal du programme intitulé CRYPT, qui effectue le cryptage d'un mot donnée en utilisant le principe suivant :

- Permuter le premier caractère du mot avec le dernier.
- Modifier l'élément milieu du mot par son ordre dans le code ASCII.

**NB. :** On suppose que le mot est une chaîne de caractères dont la taille est supérieure à trois.

## Analyse

Nom = CRYPT		
S	L.D.E.	O.U.
11	<b>Résultat</b> = Ecrire (ch )	ch
10	Insère (c,ch,p)	p
9	Efface(ch,p,1)	c
8	Convch( ORD(Ch[p]),c)	l
7	$p \leftarrow (1+l) \text{ DIV } 2$	aux
6	Insère ( aux, ch, 1)	
5	Efface(ch,1,1)	
4	$ch[l] \leftarrow ch[1]$	
3	$aux \leftarrow \text{sous-chaine}(ch,l,1)$	
2	$l \leftarrow \text{long}(ch)$	
1	ch = Donnée ("Entrer une chaîne de caractères: ")	
12	<b>Fin CRYPT</b>	

## Tableaux de déclaration des objets

Objet	Type / Nature	Rôle
ch	Chaîne de Caractères	La chaîne saisie
p	Entier	La position du milieu de la chaîne
c	Chaîne de Caractères	Sauvegarde l'ordre du l'élément milieu de la chaîne dans le code ASCII
l	Entier	La longueur de la chaîne
aux	Chaîne de Caractères	Sauvegarde le premier caractère de la chaîne

## Algorithme

0) Début CRYPT

1) Ecrire ("Entrer une chaîne de caractères: "), Lire(ch)

2)  $l \leftarrow \text{long}(ch)$

3)  $aux \leftarrow \text{sous-chaine}(ch,l,1)$

4)  $ch[l] \leftarrow ch[1]$

5) Efface(ch,1,1)

6) Insère (aux,ch, ",1)

7)  $p \leftarrow (1+l) \text{ DIV } 2$

8) Convch( ORD(Ch[p]),c)

9) Efface(ch,p,1)

10) Insère (c,ch,p)

11) Ecrire (ch )

12) Fin CRYPT

## Traduction en Pascal

```

PROGRAM CRYPT;
USES WINCRT;
VAR  ch,aux,c: STRING;
     l,p:INTEGER;
BEGIN
  WRITE ('Entrer une chaîne de caractères: ');
  READLN(ch);
  l:=length(ch);
  aux:=copy(ch,l,l);
  ch[l]:= ch[l];
  Delete(ch,l,l);
  Insert(aux,ch,l);
  P:= (l+1) DIV 2;
  Str( Ord(Ch[p]),c);
  Delete(ch,p,l);
  Insert(c,ch,p);
  WRITE (ch);
END.

```

### Un cas d'exécution

```

Entrer une chaîne de caractères: info
o110fi

```

## Retenons

- L'action d'affectation consiste à attribuer une valeur à une variable.
- L'instruction d'affectation se note avec le symbole "←".
- Une variable à affecter située à gauche de l'affectation doit avoir le même type ou un type compatible avec celui du résultat de l'évaluation de l'expression placée à droite du symbole d'affectation.

## Exercices

### Exercice 1

Soit la séquence d'instructions suivante :

```

X ← -5
X ← X*X
Y ← -X-3
Z ← (-X-Y)*3
X ← -(X+Y)*2+Z
Y ← Z*X*Y
Y ← -(Z+Y)
X ← X+Y-Z
Y ← X+Z
X ← (Y+Z)/(X/10)
Y ← ((X*Z)/Y)*9

```

Dresser la trace de la séquence algorithmique précédente pour déterminer les valeurs des variables X, Y et Z après l'exécution de la séquence d'instructions ci-dessus.

### Exercice 2

- 1) Faire une analyse, écrire un algorithme puis la traduction en Pascal du programme intitulé PERMUT qui permet de permuter les contenus de deux variables X et Y en utilisant une variable auxiliaire.
- 2) Donner une deuxième méthode permettant de résoudre cet exercice sans l'utilisation d'une variable auxiliaire.

### Exercice 3

Soit la séquence d'instructions suivante :

- 1)  $X \leftarrow 19.7$
- 2)  $Y \leftarrow 114$
- 3)  $K \leftarrow (X + Y - \text{ABS}(X-Y))/2$
- 4)  $L \leftarrow (X + Y + \text{ABS}(X-Y))/2$

- 1) Dresser la trace de la séquence algorithmique précédente pour déterminer les valeurs des variables K et L après l'exécution des instructions précédentes.
- 2) Quel est le rôle des instructions 3 et 4 ?

### Exercice 4

Écrire un algorithme qui saisit un temps en seconde puis le convertit en jours, heure, minutes et secondes.

**Exercice 5**

On vous demande d'écrire un programme en Pascal qui permet de convertir une mesure d'énergie en Joule (J) saisie au clavier en son équivalent Decijoule (dJ), Hectojoule (hJ), Kilojoule (kJ), Calorie (cal), Kilocalorie (kcal), Wattheure (W/h), Kilowattheure (kWh), BTU.

Sachant que :

- 1 calorie = 4,1855 joules
- 1 kWh = 3 600 000 J
- 1 calorie  $\approx$  3,968 321  $\times$  10<sup>-3</sup> (BTU) British Thermal Unit

**Exercice 6**

Soit A un tableaux de 3 chaînes de caractères et B un tableaux de 2 entiers.

Soit la séquence d'affectation suivante :

- A[1]  $\leftarrow$  sous-chaine("communication",1,3)
- A[2]  $\leftarrow$  concat("sports", ".", A[1])
- B[1]  $\leftarrow$  pos ("o", A[2])
- convch(2007, ch)
- A[3]  $\leftarrow$  "www."+ A[2]
- insérer(ch,A[3], 11)
- B[2]  $\leftarrow$  long(A[1] )

**Questions:**

- 1) Quel est le contenu de chaque élément des deux tableaux A et B ?
- 2) Traduire cet algorithme en Pascal.

**Exercice 7**

On se propose d'écrire un programme en Pascal intitulé POIDS qui calcule le poids d'une image numérique prise par un appareil photo numérique de sept mégapixel de résolution et d'un codage de 6 octets (48 bits/pixe).

**NB.** Le poids d'une image est le produit du nombre de pixels de l'image par le nombre d'octets par pixel.

**Exercice 8**

Écrire un programme en Pascal qui permet de calculer la vitesse de rotation de la terre autour du soleil qui est exprimée en km/s.

- NB.**
- La distance moyenne Terre-Soleil est de l'ordre de 150 000 000 km.
  - La vitesse de rotation de la terre autour du soleil = corconférence de l'orbite/ (365 jours\*24heures\*3600 secondes).
  - La corconférence de l'orbite=  $2*\pi$  \* distance moyenne Terre-Soleil

## Leçon 2

# Les opérations d'entrée / sortie

### Objectifs spécifiques

- Savoir faire une lecture et une écriture de données.
- Présenter les solutions sous forme d'un algorithme puis d'un programme.

### Plan de la leçon

#### I. Les opérations de sortie

- I.1. Définition
- I.2. Vocabulaire et syntaxe
- I.3. Formatage de l'affichage des résultats

#### II. Les opérations d'entrée

- II.1. Définition
- II.2. Vocabulaire et syntaxe

### Retenons

### Exercices

## Leçon 2

## Les opérations d'entree / sortie

*Conduire par ordre mes pensées, en commençant par les objets les plus simples et les plus aisés à connaître, pour monter peu à peu, comme par degrés, jusqu'à la connaissance des plus composés.*

DESCARTES

Nous avons vu dans la leçon précédente qu'une opération d'affectation consiste à attribuer une valeur à une variable. Cette opération se fait au niveau de la mémoire et il est intéressant de visionner le résultat de ces opérations. Nous verrons aussi comment permettre à l'utilisateur de saisir des données à mettre dans des variables par l'intermédiaire du clavier. Ces deux actions sont convenues élémentaires et s'intitulent respectivement sortie de données et entrée de données.

## I. L'opération de sortie

## Activité 1

Effectuer une analyse, écrire un algorithme et faire la traduction en Pascal du programme intitulé DISTANCE qui convertit une distance  $n = 24$  pouces en son équivalent pieds et mètres. Afficher les résultats.

On rappelle que :

- 1 pied = 12 pouces
- 1 pouce = 2.54 cm

## Analyse

Nom = DISTANCE		
S	L.D.E.	O.U.
4	Résultat = Ecrire("La mesure en mètre vaut : ",m) Ecrire("La mesure en pied vaut : ",p)	m p
3	$m \leftarrow (n * 2.54) / 100$	n
2	$p \leftarrow n \text{ div } 12$	
1	$n \leftarrow 24$	
5	Fin DISTANCE	

## Tableau de déclaration des objets

Objet	Type/nature	Rôle
m	Réel	Mesure en mètre
p	Réel	Mesure en pied
n	Entier	Mesure en pouce

## Algorithme

- 0) Début DISTANCE
- 1)  $n \leftarrow 24$
- 2)  $p \leftarrow \text{div } 12$
- 3)  $m \leftarrow (n * 2.54) / 100$
- 4) *Ecrire*("La mesure en mètre vaut : ",m)  
*Ecrire*("La mesure en pied vaut : ",p)
- 5) Fin DISTANCE

## Traduction en Pascal

```

PROGRAM DISTANCE;
USES WINCRT;
VAR
  m, p: REAL;
  n: INTEGER;
BEGIN
  n := 24 ;
  p := n DIV 12;
  m := (n * 2.54) / 100;
  WRITELN('La mesure en pied vaut : ',p);
  WRITELN('La mesure en mètre vaut : ',m);
END.

```

## I.1 Définition

La sortie de données est l'action convenue élémentaire qui consiste à écrire une donnée sur un périphérique de sortie tel que l'écran, l'imprimante, etc.

## Activité 2

Effectuer une analyse, écrire un algorithme et faire la traduction en Pascal du programme intitulé INVERSE qui saisit un entier de trois chiffres puis détermine le nombre correspondant lorsqu'on l'écrit à l'envers.

**Exemple :** 123 devient 321  
et 120 devient 21

## Analyse

Nom = INVERSE		
S	L.D.E.	O.U.
6	Résultat = Ecrire (I)	I
5	$I \leftarrow C+(D*10)+(U*100)$	C
4	$U \leftarrow E \text{ MOD } 10$	D
3	$D \leftarrow (E \text{ DIV } 10) \text{ MOD } 10$	U
2	$C \leftarrow E \text{ DIV } 100$	E
1	E = Donnée("Donner un entier naturel de trois chiffres ")	
7	Fin INVERSE	

## Tableau de déclaration des objets

Objet	Type/nature	Rôle
I	Entier	L'inverse de E
C	Entier	Chiffre des centaines de E
D	Entier	Chiffre des dizaines de E
U	Entier	Chiffre des unités de E
E	Entier	L'entier E

## Algorithme

- 0) Début INVERSE
- 1) Lire (E)
- 2)  $C \leftarrow E \text{ DIV } 100$
- 3)  $D \leftarrow (E \text{ DIV } 100) \text{ MOD } 10$
- 4)  $U \leftarrow E \text{ MOD } 10$
- 5)  $I \leftarrow C+(D*10)+(U*100)$
- 6) Ecrire (I)
- 7) Fin INVERSE

## Traduction en Pascal

```

PROGRAM INVERSE;
USES WINCRT;
VAR
    E,U,D,C,M,I : INTEGER;
BEGIN
    WRITE( 'Donner un entier naturel de trois chiffres : ' ); READLN(E);
    C:= (E DIV 100);
    D:= (E DIV 10) MOD 10;
    U:= E MOD 10;
    I:= C+(D*10)+(U*100);
    WRITE (I);
END.

```

## Un cas d'exécution

Donner un entier naturel de trois chiffres : 236  
632

## I.2 Vocabulaire et syntaxe

Au niveau de l'analyse et de l'algorithme	Au niveau de Pascal	Commentaire
Ecrire(nom_variable)	<b>WRITE</b> (nom_variable); <b>WRITELN</b> (nom_variable);	Affichage du contenu de la variable identifiée par nom_variable
Ecrire("message")	<b>WRITELN</b> ('message') ;	Affichage de la chaîne de caractères "message" telle que vous l'avez écrite.
Ecrire("message",nom_variable)	<b>WRITELN</b> ('message', nom_variable) ;	La valeur de la variable X devient égale à la valeur de la variable Y.
Ecrire (Expression)	<b>WRITELN</b> (expression);	La valeur de la variable L devient égale à la longueur de la chaîne Ch divisée par deux.

### Exemple 1 :

Analyse et algorithme
Ecrire(long*larg)
Ecrire("la surface d'un rectangle est : ", long*larg)
Ecrire("la surface d'un rectangle est : ", long , "*" , larg , "= ", s)

### Exemple 2 :

Analyse et algorithme	Pascal
Ecrire (age)	<b>WRITE</b> (age);
Ecrire (Nom," ",prénom)	<b>WRITELN</b> (Nom,' ',prénom);
Ecrire (Nom," ",prénom," a pour age : ",age)	<b>WRITELN</b> (Nom,' ',prénom,' a pour age : ' ,age);

### Remarques :

- En Pascal l'affichage d'un texte est placé entre apostrophes.
- La procédure **WRITE** affiche et laisse le curseur juste après le dernier caractère affiché.
- La procédure **WRITELN** provoque un retour à la ligne après l'affichage.

## I.3 Formatage de l'affichage des résultats

### I.3.1. Introduction

Revenons à l'activité 1 et observons les résultats obtenus.

Affichage des données	Affichage des résultats
- Mesure en pouce est 1000	- Mesure en pied vaut 8.3000000000E+01 - Mesure en mètre vaut 2.5400000000E+01
<pre> <b>Entrer une mesure en pouce: 1000</b> <b>La mesure en pied vaut : 8.3000000000E+01</b> <b>La mesure en mètre vaut : 2.5400000000E+01</b> </pre>	

Dans le langage Pascal, l'écran est par défaut, divisé en colonnes de largeur de 10 caractères, l'affichage des nombres se fait à partir de la droite de la colonne en cours. Cependant, l'utilisateur pourra imposer son format d'écriture en utilisant des facilités offertes par le langage. On pourra fixer la largeur de la colonne ainsi que le nombre de décimales s'il s'agit de l'écriture d'un réel.

Pour chacun des activités suivantes, interpréter les différents affichages et écrire le programme en Pascal en effectuant vous-même les modifications.

### I.3.2. Affichage des entiers

#### Activité 3

```

PROGRAM Format1;
USES WinCrt;
CONST Entier1 = 114;
      Entier2 = -9999999;
BEGIN
  WRITELN ('...Test sur les entiers...');
  WRITELN ('123456789012345678901234567890');
  WRITELN ('-----');
  WRITELN ( Entier1:2);
  WRITELN ( Entier1:3);
  WRITELN ( Entier1:5);
  WRITELN ( Entier1:24, Entier2);
END.

```

**L'exécution donne :**

```

...Test sur les entiers...
123456789012345678901234567890
-----
114
114
  114
                                114-9999999

```

**Interprétation :**

- L'instruction **WRITELN** (Entier1:5) permet d'afficher la valeur de l'entier Entier1 sur 5 caractères.
- Lorsque le nombre de caractères que l'on précise pour l'affichage est supérieur ou égal à ce qui est requis, la valeur est affichée en étant justifiée à droite.
- Si on fournit un nombre de caractères inférieur à ce qui est requis pour l'affichage alors le compilateur n'en tient pas compte et affiche le résultat sur un nombre correct de caractères.

**Syntaxe :**

- **WRITE** (valeur\_entière : n) affiche la valeur entière dans une colonne de n caractères à partir de la droite. Si la valeur entière comporte plus que n chiffres alors l'affichage commence par la gauche.

**I.3.3. Affichage des réels****Activité 4**

```

PROGRAM Format2;
USES WinCrt;
CONST Reel1    = 19.0;
      Reel2    = 1.23456E-3;

BEGIN
  WRITELN ('...Test sur les réels...');
  WRITELN ('123456789012345678901234567890');
  WRITELN ('-----');
  WRITELN ( Reel1);
  WRITELN ( Reel1:5:0);
  WRITELN ( Reel1:2:3);
  WRITELN ( Reel1:7:1);
  WRITELN ('-----');
  WRITELN ( Reel2);
  WRITELN ( Reel2:10:5);
  WRITELN ( Reel2:2:5);
END.

```

**L'exécution donne :**

```

...Test sur les réels...
123456789012345678901234567890
-----
 1.90000000000E+01
   19
19.000
  19.0
-----
 1.23456000000E-03
   0.00123
 0.00123

```

**Interprétation :**

- Pour les nombres réels, la syntaxe est étendue puisque nous voyons apparaître un second ":" suivi d'un nombre. La première séquence a la même signification qu'avec les entiers ; par exemple, **WRITELN** (Réal1 :5 :0) ; provoquera l'affichage de ce nombre sur 5 caractères (si c'est possible).
- La seconde séquence précise le nombre de décimales que nous désirons afficher pour le nombre en question.
- Si vous ne précisez pas le nombre de décimales alors le compilateur optera pour la notation scientifique en puissance de 10.

**Syntaxe :**

- **WRITE** (valeur\_réelle) affiche le nombre en notation scientifique (x.xxxxxE+xx précédé d'un espacement.)
- **WRITE** (valeur\_réelle : np) affiche le nombre en notation scientifique sur np positions précédé d'un espacement.
- **WRITE** (valeur\_réelle : np : nd) affiche le nombre sur np positions avec nd décimales.

**I.3.4. Affichage des chaînes de caractères****Activité 5**

```

PROGRAM Format3;
USES WinCrt;
CONST   Ch1 = 'Pêche';
        Ch2 = 'sous-marine' ;

```

```

BEGIN
  WRITELN ('...Test sur les chaines de caracteres...');
  WRITELN ('123456789012345678901234567890');
  WRITELN ('-----');
  WRITELN ( Ch1);
  WRITELN ( Ch1:5);
  WRITELN ( Ch1:7);
  WRITELN ( Ch1,Ch2);
  WRITELN ( Ch1:7,Ch2:4);
  WRITELN ( Ch1:7,Ch2:12);
END.

```

L'exécution donne :

```

...Test sur les chaines de caracteres...
123456789012345678901234567890
-----
Pêche
Pêche
  Pêche
Pêchesous-marine
  Pêchesous-marine
    Pêche sous-marine

```

**Syntaxe :**

L'affichage d'une chaîne de caractères se fait normalement à la place du curseur.

- **WRITE** (chaîne : n) affiche la chaîne sur n positions : insertion d'espacement à gauche de la chaîne si il y a moins de n caractères sinon si n est insuffisant alors ajustement automatique.

### I.3.5. Affichage des caractères

#### Activité 6

```

PROGRAM Format4;
USES WinCrt;
CONST   Char1   = 'X';
        Char2   = 'Y';

BEGIN
  WRITELN ('...Test sur les caracteres...');
  WRITELN ('123456789012345678901234567890')

```

```

WRITELN ( '-----' );
WRITELN ( Char1);
WRITELN ( Char1:2);
WRITELN ( Char1:3, Char2:4);
WRITELN ( Char1:3, Char2:5);
END.

```

L'exécution donne :

```

...Test sur les caracteres...
123456789012345678901234567890
-----
X
 X
  X  Y
   X  Y

```

**Syntaxe :**

L'affichage d'un caractère se fait normalement à la place du curseur.

- **WRITE** (car : n) affiche le caractère à la position n et insertion d'espacement à gauche du caractère.

**Remarque :**

Formater les sorties signifie qu'on désire leur imposer un format d'affichage.

### Activité 7

Écrire en Pascal le programme intitulé ConversionDinarEuro, qui convertit un montant en dinars en son équivalent en Euro et inversement.

**Traduction en Pascal**

```

PROGRAM ConversionDinarEuro;
USES WinCrt;
CONST
  Affichage = 8; (* nombres affiches sur 8 caractères *)
  Precision = 3; (* 2 décimales *)
  Change     = 1.705;
VAR
  PrixDinar, PrixEuro : real;
BEGIN
  WRITE ('Entrez un prix en Dinars : ');
  READLN (PrixDinar);
  WRITE ('Entrez un prix en Euros : ');

```

```

READLN (PrixEuro);
WRITE (PrixDinar:Affichage:Precision);
WRITELN ('D = ', (PrixDinar/Change):Affichage:Precision, 'E');
WRITE (PrixEuro:Affichage:Precision);
WRITELN ('E = ', (PrixEuro*Change):Affichage:Precision, 'D');
END.

```

## II. Les opérations d'entrée

### Activité 8

Un son numérique est caractérisé par :

- la qualité. Elle est déterminée par la fréquence d'échantillonnage c-à-d plus la fréquence est élevée, plus on prélève d'échantillons et par conséquent, meilleure sera la qualité.
- la résolution. Elle est déterminée par la plage de valeurs que peuvent prendre les échantillons, cette plage est caractérisée par le nombre de bits alloués. Plus on alloue de bits, plus la plage est grande et plus l'information sur le son est abondante.
- Le nombre de voies (monophonique, stéréophonique ...).

On vous demande de déterminer et d'afficher la taille occupée par un morceau de musique ayant les caractéristiques suivantes :

- Durée : 140 secondes
- Taux d'échantillonnage : 22000 hertz
- Résolution : 16 bits
- Son stéréophonique

- 1) Effectuer une analyse du problème intitulé MUSIQUE.
- 2) Dédire l'algorithme de cette analyse.
- 3) Traduire en Pascal cet algorithme.

### Analyse

Nom = MUSIQUE		
S	L.D.E.	O.U.
6	<b>Résultat</b> = Ecrire ("la taille de cette musique en MO est : ",T)	T
5	T ← F*R*V*D DIV (1024*1024*8)	V
4	V = Donnée ("Entrer le nombre de voies: ")	R
3	R = Donnée ("Entrer la résolution en bits: ")	F
2	F = Donnée ("Entrer la fréquence d'échantillonnage en HZ: ")	D
1	D = Donnée ("Entrer la durée en secondes : ")	
7	<b>Fin MUSIQUE</b>	

## Tableau de déclaration des objets

Objet	Type/nature	Rôle
T	Entier	La taille de la musique
V	Entier	Le nombre de voies
R	Entier	La résolution
F	Entier	Fréquence d'échantillonnage
D	Entier	La durée d'enregistrement

## Algorithme

0) Début *MUSIQUE*

1) Ecrire ("Entrer la durée en secondes : "), Lire(D)

2) Ecrire ("Entrer la fréquence d'échantillonnage en HZ : "), Lire(F)

3) Ecrire ("Entrer la résolution en bits : "), Lire(R)

4) Ecrire ("Entrer le nombre de voies : "), Lire(V)

5)  $T \leftarrow F * R * V * D \text{ DIV } (1024 * 1024 * 8)$

6) Ecrire ("la taille de cet musique en MO est : ", T)

7) Fin *MUSIQUE*

## Traduction en Pascal

```
PROGRAM MUSIQUE;
```

```
USES WINCRT;
```

```
VAR
```

```
  D,V,R,F,T: LONGINT;{*entier long sur 4 octets*}
```

```
BEGIN
```

```
  WRITELN('Entrer le temps d'enregistrement en secondes : ');
```

```
  READLN(D);
```

```
  WRITELN('Entrer la fréquence d'échantillonnage en HZ: ');
```

```
  READLN(F);
```

```
  WRITELN('Entrer la résolution en bits: '); READLN(R);
```

```
  WRITELN('Entrer le nombre de voies: '); READLN(V);
```

```
  T:= F*R*V*D DIV (1024*1024*8);
```

```
  WRITELN('la taille de cette musique en MO est : ',T);
```

```
END.
```

## Un cas d'exécution

```
Entrer le temps d'enregistrement en secondes :
```

```
45
```

```
Entrer la fréquence d'échantillonnage en HZ:
```

```
22000
```

```
Entrer la résolution en bits:
```

```
16
```

```
Entrer le nombre de voies:
```

```
2
```

```
la taille de cette musique en MO est : 3
```

## II. 1 Définition

**Une entrée consiste à introduire une donnée à partir d'une source d'entrée (clavier, souris, stylo optique ...). C'est une opération qui permet d'affecter à une variable en mémoire, une valeur de même type ou compatible avec celle de la variable.**

### Activité 9

On se propose d'écrire un programme intitulé IMAGE qui calcule le nombre de pixels et le poids d'une image numérique en méga octet (MO), d'une dimension (hauteur, largeur) exprimées en pouces et d'une résolution en dpi (points par pouce).

Analyser ce problème et en déduire l'algorithme et le programme Pascal correspondant.

#### NB.

- La résolution indique le nombre de points sur une unité de longueur de l'image.
- Le nombre de pixels dans une image = hauteur \* résolution \* longueur \* résolution
- L'image est codée en 24 bits/pixel ou 48 bits/pixel c'est à dire 8 ou 16 bits par canal R (rouge), V (vert) et B (bleu).
- Le nombre total d'octets dans l'image (poids) = Nombre de pixels \* nombre d'octets par pixel .

#### Exemple :

Soient les caractéristiques suivantes d'une image :

- largeur 4 pouces
- hauteur 5 pouces
- résolution 1200 dpi
- codage de 3 octets en 24 bits (1octet par canal R,V et B).
  - Le nombre de pixels dans cette image est :  $4 \cdot 5 \cdot 1200^2$
  - Le poids de cette image est :  $4 \cdot 5 \cdot 1200^2 \cdot 3$  octets

#### Analyse

Nom = IMAGE		
S	L.D.E.	O.U.
8	<b>Résultat</b> = Ecrire(n,p1)	n
7	$P1 \leftarrow p / (1024 \cdot 1024)$	p1
6	$p \leftarrow n \cdot c$	c
5	$n \leftarrow l \cdot r \cdot h \cdot r$	p r h l
4	c=DONNEE ("Entrer le codage de l'image : ")	
3	r=DONNEE("Entrer la résolution de l'image: ")	
2	h=DONNEE ("Entrer la hauteur de l'image : ")	
1	l=DONNEE("Entrer la largeur de l'image: ")	
9	<b>Fin IMAGE</b>	

## Tableau de déclaration des objets

Objet	Type/nature	Rôle
n	Entier	Nombre de pixels
pl	Entier	Poids de l'image en méga octet
c	Entier	Codage de l'image
p	Entier	Poids de l'image en octet
r	Entier	Résolution de l'image
h	Entier	Hauteur de l'image
l	Entier	Largeur de l'image

## Algorithme

- 0) Début IMAGE
- 1) Ecrire("Entrer la largeur de l'image: "); Lire(l)
- 2) Ecrire("Entrer la hauteur de l'image: "); Lire (h)
- 3) Ecrire("Entrer la résolution de l'image: "); Lire(r)
- 4) Ecrire("Entrer le codage de l'image: "); Lire(c)
- 5)  $n \leftarrow l*r*h*r$
- 6)  $p \leftarrow n*c$
- 7)  $P1 \leftarrow p / (1024*1024)$
- 8) Ecrire(n,p1)
- 9) Fin IMAGE

## Traduction en Pascal

```

PROGRAM IMAGE;
USES WINCRT;
VAR
  n,c,p,r,h,l: INTEGER;
  P1: REAL;
BEGIN
  WRITELN ('Entrer la largeur de l''image:');READLN(l);
  WRITELN ('Entrer la hauteur de l''image : '); READLN (h);
  WRITELN ('Entrer la résolution de l''image: '); READLN (r);
  WRITELN ('Entrer le codage de l''image: '); READLN (c);

  n:= l*r*h*r;
  p:= n*c;
  P1:= p / (1024*1024);
  WRITE(n,p1);
END.

```

## II. 2 Vocabulaire et syntaxe

Niveau analyse	Niveau algorithme	Niveau de Pascal
Variable = donnée	Lire (variable)	<b>READLN</b> (variable); ou <b>READ</b> (variable);
var1,var2,var3 = donnée	Lire (var1,var2,var3)	<b>READLN</b> (var1,var2,var3); ou <b>READ</b> (var1,var2,var3);

### Exemples :

Analyse	Algorithme	Pascal
X = donnée	Lire(X)	<b>READLN</b> (X);
Y= donnée ("Entrer Y")	Lire (variable)	<b>READLN</b> (variable); ou <b>READ</b> (variable);
(ch1,ch2,ch3) =donnée	Lire (var1,var2,var3)	<b>READLN</b> (var1,var2,var3); ou <b>READ</b> (var1,var2,var3);

### Remarques :

- Le langage PASCAL comprend deux procédures standards de lecture : **READ** et **READLN**.
- Les instructions **READ** et **READLN** permettent de transférer des données vers des variables situées dans la mémoire centrale à travers des périphériques d'entrée. Par défaut, le périphérique d'entrée utilisé est le clavier.
- Lorsqu'on utilise une seule instruction **READLN** pour lire plusieurs variables, on fait entrer les valeurs séparées par des virgules. Cependant, dans le cas où chaque variable est lue séparément, un retour de chariot est nécessaire, après chaque valeur entrée à partir du clavier, pour déclencher la lecture de celle-ci.
- La lecture de plusieurs variables de type Chaîne à la fois est non fonctionnelle.
- La seule différence qui existe entre **READLN** et **READ**, c'est que **READLN** commande un changement de ligne après la lecture de la liste des variables spécifiées comme paramètres. Ce changement de ligne ne prend effet qu'à la prochaine lecture occasionnée par un autre énoncé **READ** ou **READLN**.

**Activité 10**

Écrire en Pascal le programme intitulé DATE, qui initialise la date maintenue par le système d'exploitation, en utilisant la procédure SetDate (...) de la bibliothèque WinDos.

**NB.**

- La syntaxe de la procédure est : procedure SetDate(Annee, Mois, jour);
- les paramètres valides sont 1980..2099 pour Année, 1..12 pour Mois et 1..31 pour Jour. Si la date est invalide, la demande est ignorée.

**Traduction en Pascal**

```
PROGRAM DATE;
USES WinCrt,WinDos;
VAR
    a,m,j: WORD;

BEGIN
    WRITE ('Entrez l''année : ');READLN ( a );
    WRITE ('Entrez le mois : '); READLN ( m );
    WRITE ('Entrez le jour : '); READLN ( j );
    SetDate(a,m,j);
END.
```

**Remarque :**

le type WORD est le type entier non signé ; le domaine de définition est de 0 à 65535.

**Retenons**

– Pour afficher à l'écran une donnée quelconque, on utilise l'instruction élémentaire de sortie traduite au niveau de l'algorithme et de l'analyse par le verbe Ecrire et en Pascal par l'instruction WRITE ou WRITELN.

– Pour imposer un format d'affichage sur l'écran on doit formater les sorties. Cependant, l'utilisateur pourra imposer son format d'écriture en fixant la largeur de la colonne ainsi que le nombre de décimales s'il s'agit de l'écriture d'un réel.

– Pour saisir une donnée via le clavier pour la mettre dans une variable donnée, on utilise l'instruction élémentaire d'entrée exprimée par le verbe Lire ou le mot DONNÉE et en Pascal par l'instruction READ ou READLN.

## Exercices

### Exercice 1

Écrire un algorithme permettant de calculer et d'afficher la durée entre deux horaires exprimés en heures, minutes, secondes. Cette durée sera exprimée aussi en heures, minutes et secondes.

### Exercice 2

Faire une analyse, déduire un algorithme puis le traduire en un programme Pascal intitulé SOMME qui calcule la somme des  $n$  premiers entiers naturels non nuls en utilisant la formule mathématique suivante :  $1+2+\dots+n = (n*(n+1))/2$ .

### Exercice 3

$X$  et  $Y$  étant deux valeurs numériques. Écrire une analyse, un algorithme et la traduction en Pascal du programme intitulé CALCUL qui effectue la somme, le produit et la moyenne arithmétique de  $X$  et  $Y$ .

### Exercice 4

La puissance d'une installation de pompage est de 100 kW.

Le rendement de cette installation est de 75%.

Effectuer une analyse, un algorithme et la traduction en Pascal du programme intitulé POMPAGE qui permet de calculer le débit exprimé en litres d'eau par seconde?

Sachant que :

- La puissance utile de l'installation = La puissance de l'installation \* Le rendement de cette installation.
- Pour remonter un litre d'eau à la surface, il faut produire un travail  $W = \text{Force} * \text{profondeur (m)}$ .
- La force = masse d'un litre d'eau (1Kg) \* pesanteur  $G(10N /Kg)$ .
- Le débit de la pompe en litres par seconde = La puissance utile de l'installation/le travail  $W$ .

### Exercice 5

Deux trains, distants de 60 km, roulent l'un vers l'autre sur le même itinéraire.

Le train A avance à une vitesse de 70 km/h.

Le train B avance à une vitesse de 55 km/h.

Faire une analyse du programme intitulé RENCONTRE qui permet de calculer le temps où les deux trains vont se rencontrer?

### Exercice 6

Pour créer un répertoire, on utilise la procédure prédéfinis `MkDir(Path: string)` de l'unité DOS.

Écrire un programme en Pascal, qui permet de créer un répertoire ; le nom du répertoire et le chemin sont saisis au clavier

**Exemple :** `MkDir('C:\TEST');` permet de créer le répertoire 'test' sous la racine C :.

**Exercice 7**

Écrire un programme qui demande à l'utilisateur les coordonnées de deux points distincts du plan et qui affiche les coordonnées du point milieu.

**Exercice 8**

Préciser le résultat obtenu à la sortie d'un programme par les séquences d'instructions suivantes :

VAR var1, var2, var3, var4, var5 : REAL;

où :

var1 vaut 1.2

var2 vaut 0.00089

var3 vaut 45

var4 vaut 10095.095

var5 vaut 2.7182818281

Séquences d'instructions :

- |                         |                           |
|-------------------------|---------------------------|
| 1) WRITE(var1 : 8 : 4); | 6) WRITE(var1 : 6 : 4);   |
| 2) WRITE(var2 : 8 : 4); | 7) WRITE(var2 : 10 : 6);  |
| 3) WRITE(var3 : 4 : 1); | 8) WRITE(var3 : 10 : 6);  |
| 4) WRITE(var4 : 3 : 1); | 9) WRITE(var4 : 7 : 5);   |
| 5) WRITE(var5 : 3 : 1); | 10) WRITE(var5 : 10 : 4); |

**Exercice 9**

Écrire un programme en Pascal permettant de lire 2 nombres entiers positifs inférieurs à 999 et affiche à l'écran la multiplication et la division comme suit :

120 x 11 ..... = 1320	130 : 11 9 : ..... : 11 :
--------------------------------	------------------------------------

**Exercice 10**

Écrire un programme qui demande à l'utilisateur une valeur pour  $U_0$ ,  $r$  et  $n$  et qui affiche la  $n$ ème valeur de la suite arithmétique définie par  $U_0$  et  $U_{n+1} = U_n + r$ .  
(On rappelle la propriété :  $U_n = U_0 + n.r$ )

**Exercice 11**

Écrire un programme en Pascal qui saisit une capacité d'un disque dur puis en reprenant les caractéristiques ci-dessus, calculer et afficher :

- le nombre de disquettes nécessaires à une sauvegarde complète du disque dur, celui-ci étant supposé entièrement rempli.
- Si on utilise des CDROM au lieu des disquettes, donner le nombre de CD.
- Si on utilise un logiciel de compression de données avec un taux de compression 20%, quel sera le nombre de CD nécessaires à une sauvegarde complète du disque.