

## Correction TP1 - Introduction à Turbo Pascal

---

### BILAN DU TP1

- Nous avons donc vu que la structure générale d'un programme est la suivante :

```
PROGRAM nom_du_programme_sans_accent_ni_espace ;
  VAR nom_variable1, nom_variable2 :TYPE1 ;
      nom_variable3 : TYPE2 ;
BEGIN
  Instruction1 ;
  Instruction2 ;
  Instruction3 ;
END.
```

- Dans un programme, on a souvent besoin d'utiliser des **variables informatiques** (différentes des variables en mathématiques, le  $x$  de  $f(x)$ ) qui sont des cases mémoires de l'ordinateur. Chaque case est étiquetée par un nom, par exemple **a** dans l'exercice 01.3. Lorsqu'on utilise une variable dans un programme, il faut l'annoncer au début dans la rubrique **VAR** et indiquer le **type de variable**.
- Pour afficher une **chaîne de caractères** (c'est-à-dire du texte), ou bien la valeur d'une variable, on dispose des commandes **WRITE** et **WRITELN** qui ont la même syntaxe, mais la commande **WRITELN** est préférable car elle provoque le retour à la ligne du curseur, ce qui rend l'affichage plus lisible :
  - **WRITELN('texte à afficher');**  
affiche le texte à afficher à l'écran.
  - **WRITELN(a);**  
affiche la valeur de la variable **a**
  - **WRITELN(expression);**  
affiche la valeur de l'expression 'qui dépend de certaines variables du programme, comme **a/b** dans l'exercice 01.3.
  - **WRITELN('texte1', a, 'texte2', expression1, expression2, u, ...);**  
permet d'afficher plusieurs de ces choses les unes à la suite des autres.
- Pour attribuer une valeur à une variable de manière interactive, on dispose des commandes **READ** et **READLN**. Par exemple, **READLN(a,b,c);** permet d'entrer les données **a, b, c**, en séparant les valeurs par un espace et en tapant Entrée une fois toutes les valeurs écrites. L'ordinateur affecte alors ces valeurs aux variables correspondantes.
- Le fait de placer un **"READLN ;"** juste avant le **END.** permet d'afficher la fenêtre de calcul, le temps qu'on puisse lire les travaux effectués par l'ordinateur. Cela évite donc l'emploi des touches **Alt+F5**.

# CORRECTION DES EXERCICES

## Exercice 01.4

1. Les deux informations données nous permettent d'affirmer que  $a$  et  $b$  vérifient le système suivant :

$$\begin{cases} 32 &= 0.a + b \\ 212 &= 100.a + b \end{cases}$$

On résout ce système, on trouve que  $b = 32$  puis que  $a = 1,8$ .

2. Un programme possible est le suivant :

```
PROGRAM conversion_temperature ;
  VAR C :REAL ;
BEGIN
  WRITE('Donner la température en degrés Celsius :') ;
  READLN(C) ;
  WRITELN('La température est de ',1.8*C+32,' degrés Fahrenheit') ;
  READLN ;
END.
```

## Exercice 01.5

Il faut bien comprendre qu'en informatique, une variable est une **case mémoire**, le nom de la variable étant une étiquette pour désigner cette case mémoire. La valeur contenue dans cette case mémoire, donc la valeur de la variable, peut évoluer au cours du temps. Lorsqu'on fait une modification, l'ordinateur ne retient pas la valeur antérieure.

Par exemple ici, la variable  $u$  a contenu successivement les valeurs 2, puis 3, puis  $4 \times 3 - 1 = 11$ .

Lorsqu'on écrit  $u := 4*u - 1$ , il ne s'agit donc pas d'une égalité au sens mathématique : quand on lit " $u := 4*u - 1$ ", la variable  $u$  n'a pas la même valeur à gauche et à droite ; à droite  $u$  contient l'ancienne valeur (qui était 3), à gauche la nouvelle valeur (qui sera 11).

Lorsqu'on écrit  $u := 4*u - 1$ , on ordonne donc à l'ordinateur de prendre la valeur qu'il a stockée dans la case étiquetée  $u$ , de multiplier par 4 cette valeur puis d'ôter 1, et de remettre le résultat dans la variable  $u$ .