

## TP Informatique 1 - Introduction à Turbo Pascal

---

### 1 Notion d'algorithme

On a une tâche complexe à accomplir, par exemple un calcul. Un **algorithme** est une liste de tâches à faire **successivement** et qui permettent d'accomplir la tâche complexe.

Prenons un exemple concret : pour retirer de l'argent à un distributeur automatique, l'algorithme est le suivant :

- Introduire la carte bancaire ;
- Composer le code confidentiel ;
- Indiquer le montant du retrait ;
- Si le retrait est autorisé,
  - Retirer la carte bancaire ;
  - Dire si on veut un reçu ;
  - Prendre les billets et l'éventuel reçu.
- Sinon, reprendre votre carte.

Prenons un exemple mathématique : on souhaite que l'ordinateur résolve les équations du second degré à coefficients réels :  $ax^2 + bx + c = 0$ . Il faut lui donner des instructions, c'est-à-dire un algorithme à suivre :

- Entrer les coefficients  $a, b, c$  ;
- Calculer le discriminant  $\Delta$  ;
- Tester le signe de  $\Delta$  :
  - Si  $\Delta < 0$ , afficher "L'équation n'a pas de solution réelle" ;
  - Si  $\Delta \geq 0$ , afficher "Il y a deux solutions (éventuellement confondues),  $\frac{-b + \sqrt{\Delta}}{2a}$  et  $\frac{-b - \sqrt{\Delta}}{2a}$ ".

Malheureusement, cette liste d'instructions est incompréhensible pour votre ordinateur ! C'est pour cela qu'on utilise des langages de programmation. Nous allons apprendre le langage **PASCAL**.

Par exemple, en langage PASCAL, la liste d'instructions ci-dessus s'écrit :

```
PROGRAM equationduseconddegre ;
  VAR a,b,c,Delta :REAL ;
BEGIN
  WRITELN('Donner les coefficients a,b,c') ;
  READ(a,b,c) ;
  Delta :=b*b-4*a*c ;
  IF Delta<0 THEN
    WRITELN('L''équation n''a pas de racine réelle')
  ELSE WRITELN('L''équation a deux racines éventuellement
    confondues, '(-b+SQRT(Delta))/(2*a), 'et', (-b-SQRT(Delta))/(2*a)) ;
END.
```

Le problème est que l'ordinateur ne comprend pas non plus ce langage ! En fait, il ne comprend que le "langage machine". On utilise donc **Turbo Pascal**, c'est ce qu'on appelle un compilateur. Avec Turbo-Pascal, on écrit l'algorithme ci-dessus traduit en langage PASCAL (c'est ce qui s'appelle le **programme**), puis on **compile**, c'est-à-dire qu'on appuie sur une touche et Turbo-Pascal traduit le programme en langage machine. Après quoi, l'ordinateur a compris l'algorithme que vous vouliez lui donner ; pour résoudre une équation particulière, par exemple  $2x^2 - 7x + 3 = 0$ , il ne reste plus qu'à **exécuter** le programme. L'ordinateur va nous demander les valeurs des coefficients, ici 2, -7 et 3 ; il affichera alors le résultat adéquat.

Lorsqu'on veut faire accomplir une tâche à l'ordinateur, il y a donc plusieurs étapes pour vous :

- étape 1 : ANALYSE DE L'ALGORITHME. Au brouillon, on étudie le problème, on identifie les différentes étapes à suivre, bref on essaye de trouver un algorithme pour répondre au problème.
- étape 2 : PROGRAMMATION. On écrit le programme en langage PASCAL.
- étape 3 : on compile.
- étape 4 : on exécute.

L'étape la plus importante est la première ! En effet, l'écriture du programme est une simple traduction de l'algorithme. Si vous avez réussi à trouver un bon algorithme, la seule difficulté est de ne pas faire de faute de frappe, de vocabulaire ou de syntaxe.

## 2 Afficher du texte, un résultat, entrée une donnée

### Exercice 01.1

1. Taper le programme suivant, le compiler et l'exécuter :

```
PROGRAM bonjour ;
BEGIN
    WRITELN('bonjour');
END.
```

(Pour voir l'écran, appuyer sur ALT+F5).

2. L'exécuter une seconde fois.
3. Remplacer WRITELN par WRITE, compiler et exécuter deux fois. Quelle est la différence entre WRITE et WRITELN ?
4. Remplacer 'bonjour' par 'il fait beau aujourd'hui'. Que se passe-t-il ? Comment faire pour afficher la phrase en entier ?

### Exercice 01.2

1. Taper le programme suivant, le compiler et l'exécuter :

```
PROGRAM age ;
    VAR a :INTEGER ;
BEGIN
    WRITELN('Donnez votre âge :') ;
    READLN(a) ;
    WRITELN('Vous avez ',a,' ans') ;
END.
```

2. Rajouter, juste avant le END., une ligne avec "READLN ;". Que se passe-t-il ? Quel est l'intérêt de faire cela ?
3. A la première ligne, remplacer age par âge, puis par mon age, puis par mon\_age ; que se passe-t-il dans chaque cas ?
4. Que se passe-t-il si on supprime la ligne commençant par VAR ?

### Exercice 01.3

1. Taper le programme suivant, le compiler et l'exécuter :

```
PROGRAM division ;
  VAR a,b :REAL ;
BEGIN
  WRITELN('Donner le nombre a :') ;
  READLN(a) ;
  WRITELN('Donner le nombre b :') ;
  READLN(b) ;
  WRITELN('Le quotient de ces nombres est', a/b) ;
  READLN ;
END.
```

2. Une autre possibilité pour faire la même chose est le programme suivant ; tapez le :

```
PROGRAM division ;
  VAR a,b :REAL ;
BEGIN
  WRITELN('Donner les nombres a et b :') ;
  READLN(a,b) ;
  WRITELN('Le quotient de ces nombres est',a/b) ;
  READLN ;
END.
```

Comment doit-on taper les nombres  $a$  et  $b$  pour que cela fonctionne ?

### Exercice 01.4

1.  $C$  désigne la température expérimentée en degrés Celsius, et  $F$  la température exprimée en degrés Fahrenheit. On sait que  $F$  dépend linéairement de  $C$ , c'est-à-dire que  $F = aC + b$ . Déterminer  $a$  et  $b$  sachant que l'eau gèle à 32 degrés Fahrenheit et bout à 212 degrés Fahrenheit.
2. Ecrire un programme qui demande la température en degrés Celsius et qui affiche la température en degrés Fahrenheit.

## 3 Affectation d'une variable

Nous avons vu que pour donner une valeur à une variable  $u$ , l'utilisateur peut le faire lui-même à l'aide de la commande READLN. On peut aussi écrire dans le programme  $u:=1$  ; pour affecter la valeur 1 à la variable  $u$ . Si  $v$  est une autre variable à laquelle on a déjà affecté une valeur, on peut aussi écrire  $u :=2*v$  ;.

### Exercice 01.5

Taper, compiler et exécuter le programme suivant, et commenter :

```
PROGRAM affectation ;
  VAR u :REAL ;
BEGIN
  u :=2;
  WRITELN(u) ;
  u:=3 ;
  WRITELN(u) ;
  u :=4*u-1 ;
  WRITELN(u) ;
```

```
    READLN ;  
END.
```

### Exercice 01.6

1. Déterminer les valeurs contenues par les variables  $x$  et  $y$  à la fin de l'exécution de ce programme :

```
PROGRAM melange ;  
    VAR x,y :REAL ;  
BEGIN  
    WRITE('Donner la valeur de x :') ;  
    READLN(x) ;  
    WRITE('Donner la valeur de y :') ;  
    READLN(y) ;  
    x := x+y ;  
    y := x-y ;  
    x := x-y ;  
END.
```

2. Vérifier en exécutant le programme suivant (le même sauf qu'il affiche les valeurs successives prises par  $x$  et  $y$ ).

```
PROGRAM melange ;  
    VAR x,y :REAL ;  
BEGIN  
    WRITE('Donner la valeur de x :') ;  
    READLN(x) ;  
    WRITE('Donner la valeur de y :') ;  
    READLN(y) ;  
    x := x+y ;  
    WRITELN('après la première modification, x vaut ',x,' et y vaut ',y) ;  
    y := x-y ;  
    WRITELN('après la seconde modification, x vaut ',x,' et y vaut ',y) ;  
    x := x-y ;  
    WRITELN('après la troisième et dernière modification, x vaut ',x,' et y vaut ',y) ;  
    READLN ;  
END.
```

3. Que se passe-t-il lorsqu'on échange les deux dernières affectations  $x := x-y$  et  $y := x-y$ ? Quelles sont les valeurs finales de  $x$  et de  $y$ ?

---

## 4 Bilan, correction, téléchargement de Turbo Pascal

Tout est sur la page <http://blparc.fr>  
(en bas de la page, cliquer sur "Informatique ECS1").