

Rappel : ce cours d'algorithmique et de programmation
est enseigné à l'Université Paris 7,
dans la spécialité PISE du **Master MECI** (ancien DESS AIGES)
par [Christophe Darmangeat](#)

Page d'Accueil

PARTIE 4

CORRIGÉS DES EXERCICES

Exercice 4.1

Aucune difficulté, il suffit d'appliquer la règle de la transformation du OU en ET vue en cours (loi de Morgan). Attention toutefois à la rigueur dans la transformation des conditions en leur contraire...

```
Si Tutu <= Toto + 4 ET Tata <> "OK" Alors  
  Tutu ← Tutu - 1  
Sinon  
  Tutu ← Tutu + 1  
Finsi
```

[énoncé](#) - [retour au cours](#)

Exercice 4.2

```
Variables h, m en Numérique  
Début  
Ecrire "Entrez les heures, puis les minutes : "  
Lire h, m  
m ← m + 1  
Si m = 60 Alors  
  m ← 0  
  h ← h + 1  
FinSi  
Si h = 24 Alors  
  h ← 0  
FinSi  
Ecrire "Dans une minute il sera ", h, "heure(s) ", m, "minute(s)"  
Fin
```

[énoncé](#) - [retour au cours](#)

Exercice 4.3

```
Variables h, m, s en Numérique  
Début
```

```
Ecrire "Entrez les heures, puis les minutes, puis les secondes : "  
Lire h, m, s  
s ← s + 1  
Si s = 60 Alors  
    s ← 0  
    m ← m + 1  
FinSi  
Si m = 60 Alors  
    m ← 0  
    h ← h + 1  
FinSi  
Si h = 24 Alors  
    h ← 0  
FinSi  
Ecrire "Dans une seconde il sera ", h, "h", m, "m et ", s, "s"  
Fin
```

[énoncé - retour au cours](#)

Exercice 4.4

```
Variables n, p en Numérique  
Début  
Ecrire "Nombre de photocopies : "  
Lire n  
Si n ≤ 10 Alors  
    p ← n * 0,1  
SinonSi n ≤ 30 Alors  
    p ← 10 * 0,1 + (n - 10) * 0,09  
Sinon  
    p ← 10 * 0,1 + 20 * 0,09 + (n - 30) * 0,08  
FinSi  
Ecrire "Le prix total est: ", p  
Fin
```

[énoncé - retour au cours](#)

Exercice 4.5

```
Variable sex en Caractère  
Variable age en Numérique  
Variables C1, C2 en Booléen  
Début  
Ecrire "Entrez le sexe (M/F) : "  
Lire sex  
Ecrire "Entrez l'âge: "  
Lire age  
C1 ← sex = "M" ET age > 20  
C2 ← sex = "F" ET (age > 18 ET age < 35)  
Si C1 ou C2 Alors  
    Ecrire "Imposable"
```

```
Sinon
  Ecrire "Non Imposable"
FinSi
Fin
```

[énoncé - retour au cours](#)

Exercice 4.6

Cet exercice, du pur point de vue algorithmique, n'est pas très méchant. En revanche, il représente dignement la catégorie des énoncés piégés.

En effet, rien de plus facile que d'écrire : si le candidat a plus de 50%, il est élu, sinon s'il a plus de 12,5 %, il est au deuxième tour, sinon il est éliminé. Hé hé hé... mais il ne faut pas oublier que le candidat peut très bien avoir eu 20 % mais être tout de même éliminé, tout simplement parce que l'un des autres a fait plus de 50 % et donc qu'il n'y a pas de deuxième tour !...

Moralité : ne jamais se jeter sur la programmation avant d'avoir soigneusement mené l'analyse du problème à traiter.

```
Variables A, B, C, D en Numérique
Variables C1, C2, C3, C4 en Booléen
Début
  Ecrire "Entrez les scores des quatre prétendants :"
  Lire A, B, C, D
  C1 ← A > 50
  C2 ← B > 50 ou C > 50 ou D > 50
  C3 ← A >= B et A >= C et A >= D
  C4 ← A >= 12,5
  Si C1 Alors
    Ecrire "Élu au premier tour"
  SinonSi C2 ou Non(C4) Alors
    Ecrire "Battu, éliminé, sorti !!!"
  SinonSi C3 Alors
    Ecrire "Ballotage favorable"
  Sinon
    Ecrire "Ballotage défavorable"
  FinSi
Fin
```

[énoncé - retour au cours](#)

Exercice 4.7

Là encore, on illustre l'utilité d'une bonne analyse. Je propose deux corrigés différents. Le premier suit l'énoncé pas à pas. C'est juste, mais c'est vraiment lourd. La deuxième version s'appuie sur une vraie compréhension d'une situation pas si embrouillée qu'elle n'en a l'air.

Dans les deux cas, un recours aux variables booléennes aère sérieusement l'écriture.

Donc, premier corrigé, on suit le texte de l'énoncé pas à pas :

```
Variables age, perm, acc, assur en Numérique
Variables C1, C2, C3 en Booléen
Variable situ en Caractère
Début
```

```
Ecrire "Entrez l'âge: "  
Lire age  
Ecrire "Entrez le nombre d'années de permis: "  
Lire perm  
Ecrire "Entrez le nombre d'accidents: "  
Lire acc  
Ecrire "Entrez le nombre d'années d'assurance: "  
Lire assur  
C1 ← age >= 25  
C2 ← perm >= 2  
C3 ← assur > 5  
Si Non(C1) et Non(C2) Alors  
  Si acc = 0 Alors  
    situ ← "Rouge"  
  Sinon  
    situ ← "Refusé"  
  FinSi  
SinonSi ((Non(C1) et C2) ou (C1 et Non(C2))) Alors  
  Si acc = 0 Alors  
    situ ← "Orange"  
  SinonSi acc = 1 Alors  
    situ ← "Rouge"  
  Sinon  
    situ ← "Refusé"  
  FinSi  
Sinon  
  Si acc = 0 Alors  
    situ ← "Vert"  
  SinonSi acc = 1 Alors  
    situ ← "Orange"  
  SinonSi acc = 2 Alors  
    situ ← "Rouge"  
  Sinon  
    situ ← "Refusé"  
  FinSi  
FinSi  
Si C3 Alors  
  Si situ = "Rouge" Alors  
    situ ← "Orange"  
  SinonSi situ = "Orange" Alors  
    situ ← "Vert"  
  SinonSi situ = "Vert" Alors  
    situ ← "Bleu"  
  FinSi  
FinSi  
Ecrire "Votre situation : ", situ  
Fin
```

Vous trouvez cela compliqué ? Oh, certes oui, ça l'est ! Et d'autant plus qu'en lisant entre les lignes, on pouvait s'apercevoir que ce galimatias de tarifs recouvre en fait une logique très simple : un système à points. Et il suffit de comptabiliser les points pour que tout s'éclaire... Reprenons juste après l'affectation des trois variables booléennes C1, C2, et C3. On écrit :

```
P ← 0
Si Non(C1) Alors
  P ← P + 1
FinSi
Si Non(C2) Alors
  P ← P + 1
FinSi
P ← P + acc
Si P < 3 et C3 Alors
  P ← P - 1
FinSi
Si P = -1 Alors
  situ ← "Bleu"
SinonSi P = 0 Alors
  situ ← "Vert"
SinonSi P = 1 Alors
  situ ← "Orange"
SinonSi P = 2 Alors
  situ ← "Rouge"
Sinon
  situ ← "Refusé"
FinSi
Ecrire "Votre situation : ", situ
Fin
```

Cool, non ?

[énoncé - retour au cours](#)

Exercice 4.8

En ce qui concerne le début de cet algorithme, il n'y a aucune difficulté. C'est de la saisie bête et même pas méchante:

```
Variables J, M, A, JMax en Numérique
Variables VJ, VM, B en Booléen
Début
Ecrire "Entrez le numéro du jour"
Lire J
Ecrire "Entrez le numéro du mois"
Lire M
Ecrire "Entrez l'année"
Lire A
```

C'est évidemment ensuite que les ennuis commencent... La première manière d'aborder la chose consiste à se dire que fondamentalement, la structure logique de ce problème est très simple. Si nous créons deux variables booléennes VJ et VM, représentant respectivement la validité du jour et du mois entrés, la fin de l'algorithme sera d'une simplicité biblique (l'année est valide par définition, si on évacue le débat byzantin concernant l'existence de l'année zéro) :

```
Si VJ et VM alors
  Ecrire "La date est valide"
Sinon
```

```
Ecrire "La date n'est pas valide"
FinSi
```

Toute la difficulté consiste à affecter correctement les variables VJ et VM, selon les valeurs des variables J, M et A. Dans l'absolu, VJ et VM pourraient être les objets d'une affectation monstrueuse, avec des conditions atrocement composées. Mais franchement, écrire ces conditions en une seule fois est un travail de bénédictin sans grand intérêt. Pour éviter d'en arriver à une telle extrémité, on peut sérier la difficulté en créant deux variables supplémentaires :

B : variable booléenne qui indique s'il s'agit d'une année bissextile

JMax : variable numérique qui indiquera le dernier jour valable pour le mois entré.

Avec tout cela, on peut y aller et en ressortir vivant.

On commence par initialiser nos variables booléennes, puis on traite les années, puis les mois, puis les jours.

On note "dp" la condition "divisible par" :

```
B ← A dp 400 ou (non(A dp 100) et A dp 4)
Jmax ← 0
VM ← M >= 1 et M <= 12
Si VM Alors
  Si M = 2 et B Alors
    JMax ← 29
  SinonSi M = 2 Alors
    JMax ← 28
  SinonSi M = 4 ou M = 6 ou M = 9 ou M = 11 Alors
    JMax ← 30
  Sinon
    JMax ← 31
FinSi
VJ ← J >= 1 et J <= Jmax
FinSi
```

Cette solution a le mérite de ne pas trop compliquer la structure des tests, et notamment de ne pas répéter l'écriture finale à l'écran. Les variables booléennes intermédiaires nous épargnent des conditions composées trop lourdes, mais celles-ci restent néanmoins sérieuses.

Une approche différente consisterait à limiter les conditions composées, quitte à le payer par une structure beaucoup plus exigeante de tests imbriqués. Là encore, on évite de jouer les extrémistes et l'on s'autorise quelques conditions composées lorsque cela nous simplifie l'existence. On pourrait aussi dire que la solution précédente "part de la fin" du problème (la date est elle valide ou non ?), alors que celle qui suit "part du début" (quelles sont les données entrées au clavier ?) :

```
Si M < 1 ou M > 12 Alors
  Ecrire "Date Invalide"
SinonSi M = 2 Alors
  Si A dp 400 Alors
    Si J < 1 ou J > 29 Alors
      Ecrire "Date Invalide"
    Sinon
      Ecrire "Date Valide"
    FinSi
  SinonSi A dp 100 Alors
    Si J < 1 ou J > 28 Alors
```

```

    Ecrire "Date Invalide"
  Sinon
    Ecrire "Date Valide"
  FinSi
SinonSi A dp 4 Alors
  Si J < 1 ou J > 29 Alors
    Ecrire "Date Invalide"
  Sinon
    Ecrire "Date Valide"
  FinSi
Sinon
  Si J < 1 ou J > 28 Alors
    Ecrire "Date Invalide"
  Sinon
    Ecrire "Date Valide"
  FinSi
FinSi
SinonSi M = 4 ou M = 6 ou M = 9 ou M = 11 Alors
  Si J < 1 ou J > 30 Alors
    Ecrire "Date Invalide"
  Sinon
    Ecrire "Date Valide"
  FinSi
Sinon
  Si J < 1 ou J > 31 Alors
    Ecrire "Date Invalide"
  Sinon
    Ecrire "Date Valide"
  FinSi
FinSi

```

On voit que dans ce cas, l'alternative finale (Date valide ou invalide) se trouve répétée un grand nombre de fois. Ce n'est en soi ni une bonne, ni une mauvaise chose. C'est simplement une question de choix stylistique. Personnellement, j'avoue préférer assez nettement la première solution, qui fait ressortir beaucoup plus clairement la structure logique du problème (il n'y a qu'une seule alternative, autant que cette alternative ne soit écrite qu'une seule fois).

Il convient enfin de citer une solution très simple et élégante, un peu plus difficile peut-être à imaginer du premier coup, mais qui avec le recul apparaît comme très immédiate. Sur le fond, cela consiste à dire qu'il y a quatre cas pour qu'une date soit valide : celui d'un jour compris entre 1 et 31 dans un mois à 31 jours, celui d'un jour compris entre 1 et 30 dans un mois à 30 jours, celui d'un jour compris entre 1 et 29 en février d'une année bissextile, et celui d'un jour de février compris entre 1 et 28. Ainsi :

```

B ← (A dp 4 et Non(A dp 100)) ou A dp 400
K1 ← (m=1 ou m=3 ou m=5 ou m=7 ou m=8 ou m=10 ou m=12) et (J>=1 et J<=31)
K2 ← (m=4 ou m=6 ou m=9 ou m=11) et (J>=1 et J<=30)
K3 ← m=2 et B et J>=1 et J<=29
K4 ← m=2 et J>=1 et J<=28
Si K1 ou K2 ou K3 ou K4 Alors
  Ecrire "Date valide"
Sinon
  Ecrire "Date non valide"
FinSi
Fin

```

Tout est alors réglé avec quelques variables booléennes et quelques conditions composées, en un minimum de lignes de code.

La morale de ce long exercice - et non moins long corrigé, c'est qu'un problème de test un peu compliqué admet une pléiade de solutions justes...

...Mais que certaines sont plus astucieuses que d'autres !

[énoncé](#) - [retour au cours](#)