

FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

CORRIGES DES EXERCICES : 4.1 à 4.8

Exercice 4.1

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 \leq Toto + 4 **ET** Tata \neq "OK" **Alors**

Tutu \leftarrow Tutu - 1

Sinon

Tutu \leftarrow Tutu + 1

Finsi

Exercice 4.2

ALGORITHME Exo_4_2

Variables h, m **en Numérique**

Début

Ecrire "Entrez les heures, puis les minutes : "

Lire h, m

m \leftarrow m + 1

Si m = 60 **Alors**

m \leftarrow 0

h \leftarrow h + 1

FinSi

Si h = 24 **Alors**

h \leftarrow 0

FinSi

Ecrire "Dans une minute il sera ", h, "heure(s) ", m, "minute(s)"

Fin



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

Exercice 4.3

ALGORITHME Exo_4_3

Variables h, m, s **en Numérique**

Début

Ecrire "Entrez les heures, puis les minutes, puis les secondes : "

Lire h, m, s

$s \leftarrow s + 1$

Si $s = 60$ **Alors**

$s \leftarrow 0$

$m \leftarrow m + 1$

FinSi

Si $m = 60$ **Alors**

$m \leftarrow 0$

$h \leftarrow h + 1$

FinSi

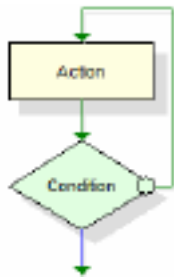
Si $h = 24$ **Alors**

$h \leftarrow 0$

FinSi

Ecrire "Dans une seconde il sera ", h, "h", m, "m et ", s, "s"

Fin



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

Exercice 4.4

ALGORITHME Exo_4_4

Variables n, p en Numérique

Début

Ecrire "Nombre de photocopies : "

Lire n

Si $n \leq 10$ **Alors**

$p \leftarrow n * 0,1$

SinonSi $n \leq 30$ **Alors**

$p \leftarrow 10 * 0,1 + (n - 10) * 0,09$

Sinon

$p \leftarrow 10 * 0,1 + 20 * 0,09 + (n - 30) * 0,08$

FinSi

Ecrire "Le prix total est: ", p

Fin



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

Exercice 4.5

ALGORITHME Exo_4_5

Variable sSexe en alphanumerique

Variable iAge en Numérique

Variables bH20, bF18_35 en Booléen

Début

Ecrire "Entrez le sexe (M/F) : "

Lire sSexe

Ecrire "Entrez l'âge: "

Lire iAge

bH20 ← sSexe == "M" ET iAge > 20

bF18_35 ← sSexe == "F" ET (iAge > 18 ET iAge < 35)

Si bH20 ou bF18_35 **Alors**

Ecrire "Imposable"

Sinon

Ecrire "Non Imposable"

FinSi

Fin



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

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. 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.

ALGORITHME Exo_4_6

Variables iScore1, iScore2, iScore3, iScore4 **en Numérique**

Variables b1erTour, bPerdu, bBalOK, bBalNOOK **en Booléen**

Début

Ecrire "Entrez les scores des quatre prétendants :"

Lire iScore1, iScore2, iScore3, iScore4

b1erTour ← iScore1 > 50

bPerdu ← iScore2 > 50 ou iScore3 > 50 ou iScore4 > 50

bBalOK ← iScore1 >= iScore2 et iScore1 >= iScore3 et iScore1 >= iScore4

bBalNOOK ← iScore1 < 12,5

Si b1erTour **Alors**

Ecrire "Elu au premier tour"

SinonSi bPerdu ou bBalNOOK **Alors**

Ecrire "Battu, éliminé, sorti !!!"

SinonSi bBalOK **Alors**

Ecrire "Ballotage favorable"

Sinon

Ecrire "Ballotage défavorable"

FinSi

Fin



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

Exercice 4.7

Là encore, on illustre l'utilité d'une bonne analyse. 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 :

ALGORITHME Exo_4_7

Variables iAge, iPerm, iAcc, iAssur **en Numerique**

Variables bAge25, bPerm2, bAssur5 **en Booleen**

Variable sSituation **en Alphanumerique**

Début

Ecrire "Entrez l'âge: "

Lire iAge

Ecrire "Entrez le nombre d'années de permis: "

Lire iPerm

Ecrire "Entrez le nombre d'accidents: "

Lire iAcc

Ecrire "Entrez le nombre d'années d'assurance: "

Lire iAssur

bAge25 ← iAge >= 25

bPerm2 ← iPerm >= 2

bAssur5 ← iAssur > 5

Si Non(bAge25) et Non(bPerm2) **Alors**

Si iAcc == 0 **Alors**

sSituation ← "Rouge"

Sinon

sSituation ← "Refusé"

FinSi

Sinonsi ((Non(bAge25) et bPerm2) ou (bAge25 et Non(bPerm2))) **Alors**

Si iAcc == 0 **Alors**



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

```

    sSituation ← "Orange"
    SinonSi iAcc == 1 Alors
        sSituation ← "Rouge"
    Sinon
        sSituation ← "Refusé"
    FinSi
Sinon
    Si iAcc == 0 Alors
        sSituation ← "Vert"
    SinonSi iAcc == 1 Alors
        sSituation ← "Orange"
    SinonSi iAcc == 2 Alors
        sSituation ← "Rouge"
    Sinon
        sSituation ← "Refusé"
    FinSi
FinSi
Si bAssur5 Alors
    Si sSituation == "Rouge" Alors
        sSituation ← "Orange"
    SinonSi sSituation == "Orange" Alors
        sSituation ← "Vert"
    SinonSi sSituation == "Vert" Alors
        sSituation ← "Bleu"
    FinSi
FinSi
Ecrire "Votre situation : ", sSituation
  
```

Fin

Long et un peu compliqué, alors qu'en fait, on peut utiliser une logique très simple : un système à points. Et il suffit de comptabiliser les points ... Reprenons juste après l'affectation des trois variables booléennes bAge25, bPerm2 et bAssur5.



FORMATION AFPA - DEVELOPPEUR LOGICIEL -

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

On écrit (en considérant le début de l'algorithme, et les bonnes déclarations de variables) :

```
iPoint ← 0
Si Non(bAge25) Alors
    iPoint ← iPoint + 1
FinSi
Si Non(bPerm2) Alors
    iPoint ← iPoint + 1
FinSi
iPoint ← iPoint + acc
Si iPoint < 3 et bAssur5 Alors
    iPoint ← iPoint - 1
FinSi
Si iPoint = -1 Alors
    sSituation ← "Bleu"
SinonSi iPoint = 0 Alors
    sSituation ← "Vert"
SinonSi iPoint = 1 Alors
    sSituation ← "Orange"
SinonSi iPoint = 2 Alors
    sSituation ← "Rouge"
Sinon
    sSituation ← "Refusé"
FinSi
Ecrire "Votre situation : ", sSituation
```

Fin



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

Exercice 4.8

En ce qui concerne le début de cet algorithme, il n'y a aucune difficulté. C'est de la saisie simple :

ALGORITHME Exo_4_8

Variables J, M, A, JMax en Numérique

Variables VJ, VM, B en Booleen

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 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 grosse affectation illisible, avec des conditions trop composées. Pour éviter d'en arriver à une telle extrémité, on peut diminuer 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.



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

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 les variables booléennes intermédiaires nous épargnent des conditions composées trop lourdes.

Une approche différente consisterait à limiter les conditions composées, quitte à le payer par une structure beaucoup plus exigeante de tests imbriqués. Là 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
  
```



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

```

    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 > 29Alors
        Ecrire "Date Invalide"
    Sinon
        Ecrire "Date Valide"
    FinSi
Sinon
    Si J < 1 ou J > 28 Alors
        Ecrire "Date Invalide"
    Sinon
        Ecrire "Date Valide"
    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

```



FORMATION AFPA - DEVELOPPEUR LOGICIEL –

(NIVEAU III)



JJP

ALGORITHME ET PSEUDO-CODE

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.

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 !