

# Initiation à l'algorithmique

(Par Thierry Dechambre)

Un **algorithme** est une suite finie et non ambiguë d'opérations ou d'instructions permettant de résoudre un problème ou d'obtenir un résultat. Il est dit correct lorsque, pour chaque instance du problème, il se termine en produisant la bonne sortie, c'est-à-dire qu'il résout le problème posé. (Wikipédia)

Un algorithme comprend des structures de données et des structures de contrôle.

## 1. Les structures de données

Elles représentent les intrants, les calculs intermédiaires et les résultats du problème. Elles peuvent être de natures différentes : numériques, alphabétiques, booléennes (vrai / faux), tableaux pouvant contenir n'importe quel type de données suscitées voire même des tableaux de tableaux ...

## 2. Les structures de contrôle

Essentiellement de deux types : les branchements et les boucles.

### a) Les branchements

#### ✓ **Branchement inconditionnel ALLER A**

Il s'agit d'un saut jusqu'à un endroit défini de l'algorithme au moyen d'un étiquette (un mot suivi de :). Exemple :

DEBUT :

Instruction 1

Instruction 2

**ALLER A SUIVANT**

Instruction 3

Instruction 4

SUIVANT :

Instruction 5

Dans cet exemple, Instruction 3 et 4 ne seront pas exécutées.

#### ✓ **Branchement conditionnel SI , SELON**

Nous ne parlerons que de SI ici.

Instruction 1

**SI <condition> ALORS**

Instruction 2

**SINON**

Instruction 3

Instruction 4

**FIN SI**

Instruction 5

Si condition est vraie Instruction 2 sera exécutée. Si condition est fausse Instruction 3 et 4 seront exécutées. Dans les deux cas, Instruction 1 et 5 seront exécutées. Evidemment vous pouvez mettre plus d'une instruction dans un SI, vous pouvez même imbriquer des SI dans des SI. A l'opposé, un SI réduit à l'extrême donne ceci :

```
SI <condition> ALORS Instruction FIN SI //instruction sera exécutée ssi condition est //vraie
```

Par convention, on utilise // pour indiquer que tout ce qui suit sur cette ligne est du commentaire.

## b) Les boucles

Essentiellement deux types de boucles.

### ✓ Boucle TANT QUE

TANT QUE <condition> FAIRE

Instruction1

Instruction2

...

InstructionN

FIN TANT QUE

Les instructions 1 à N seront répétées tant que condition sera vraie.

### ✓ Boucle POUR

POUR <nombre1> A <nombre2> FAIRE

Instruction1

Instruction2

...

InstructionN

FIN POUR

Les instructions 1 à N seront répétées ((nombre2 – nombre1)+1) fois. Exemple :

Instruction1

POUR 1 A 3 FAIRE

Instruction2

Instruction3

FIN POUR

Instruction4

Les instructions 2 et 3 seront exécutées trois fois de suite. Evidemment, si nombre2 est inférieur à nombre1, on ne rentre pas dans la boucle les instructions comprises dans la boucle ne seront pas exécutées.

En règle générale, lorsqu'on connaît ou qu'on peut calculer le nombre de tours qu'on doit faire dans la boucle, on utilise une boucle POUR, dans le cas contraire, on utilise une boucle TANT QUE.

Voilà, vous connaissez tout de l'algorithmique, ce n'était pas si compliqué☺.

Servons nous maintenant de cette connaissance toute fraîche pour résoudre un problème.

Voici l'énoncé du problème :

Convertissez n'importe quel nombre compris entre 1 et 3999 en chiffre romain.

Pour mémoire voici les chiffres romains :

I	II	III	IV	V	VI	VII	VIII	IX
1	2	3	4	5	6	7	8	9
X	XI	XII	XX	XXX	XL	L	LX	LXX
10	11	12	20	30	40	50	60	70
LXXX	XC	C	CC	CCC	CD	D	CM	M
80	90	100	200	300	400	500	900	1000

Commençons par le commencement, définissons les données dont nous allons avoir besoin.

### DONNEES

DECLARE NbSaisi COMME nombre entier

DECLARE NbRomain COMME chaîne de caractères

DECLARE R COMME nombre entier

Ces 3 variables devraient suffire. Attaquons l'algo à présent.

## CODE

DEBUT :

```
AFFICHER « Saisissez un nombre compris entre 1 et 3999 »  
LIRE NbSaisi
```

Il nous faut maintenant vérifier que le nombre saisi corresponde bien aux critères. Vous verrez dans tout bon livre sur l'algorithmique que cela se fait par une boucle TANT QUE.

```
TANT QUE NbSaisi < 1 OU NbSaisi > 3999 FAIRE  
    AFFICHER « vous devez saisir un nombre compris entre 1 et 3999 »  
    LIRE NbSaisi  
FIN TANT QUE
```

Mais l'algorithmique n'est pas un long fleuve tranquille ... Ce serait plutôt un delta, il est rare qu'il n'existe qu'un seul chemin. Ainsi, nous aurions pu faire la même chose avec :

```
SI NbSaisi > 0 ET NbSaisi < 4000 ALORS ALLER A SUIVANT FIN SI  
AFFICHER « vous devez saisir un nombre compris entre 1 et 3999 »  
ALLER A DEBUT  
SUIVANT :
```

Nous garderons néanmoins la version traditionnellement enseignée : la boucle TANT QUE.

Passons maintenant au traitement du nombre.

MILLE :

```
R = NbSaisi / 1000 // R contient le chiffre des milliers du nombre  
//saisi  
SI R=0 ALORS ALLER A CENT FIN SI //Si NbSaisi<1000 on passe aux centaines  
NbSaisi = NbSaisi - ( R x 1000 ) // On enlève le chiffre des milliers du nombre  
//saisi  
POUR 1 A R FAIRE // Traite les cas 1000, 2000, et 3000  
    NbRomain = NbRomain + M  
FIN POUR
```

Voilà, le traitement des milliers est fait, c'était le plus simple. Passons aux centaines.

CENT :

```
R = NbSaisi / 100 // R contient le chiffre des centaines du nombre  
//saisi  
SI R=0 ALORS ALLER A DIX FIN SI //Si NbSaisi<100 on passe aux dizaines  
NbSaisi = NbSaisi - ( R x 100 ) // On enlève le chiffre des centaines du nombre  
//saisi  
SI R = 9 ALORS // Traite le cas 900 et passe aux dizaines  
    NbRomain = NbRomain + CM  
    ALLER A DIX  
FIN SI  
SI R = 4 ALORS // Traite le cas 400 et passe aux dizaines  
    NbRomain = NbRomain + CD  
    ALLER A DIX  
FIN SI  
SI R >= 5 ALORS // Traite le cas 500 et prépare 600, 700 et 800  
    NbRomain = NbRomain + D
```

```

R = R - 5
FIN SI
POUR 1 A R FAIRE // Traite les cas 100, 200, 300, 600, 700 et 800
    NbRomain = NbRomain + C
FIN POUR

```

Un peu plus compliqué, n'est-ce pas ? IL a fallu traiter les cas 9, 4 et 5 qui n'étaient pas présents pour les milliers. Le cas des dizaines et des unités sera du même acabit. Ce qui nous donne l'algorithme suivant :

```

DIX :
R = NbSaisi / 10 // R contient le chiffre des dizaines du nombre
//saisi

SI R = 0 ALORS ALLER A UN FIN SI // Si NbSaisi < 10 on passe au chiffre des unités
NbSaisi = NbSaisi - ( R x 10 ) // On enlève le chiffre des dizaines de NbSaisi
SI R = 9 ALORS // Traite le cas 90 et passe au chiffre des unités
    NbRomain = NbRomain + XC
    ALLER A UN
FIN SI
SI R = 4 ALORS // Traite le cas 40 et passe au chiffre des unités
    NbRomain = NbRomain + XL
FIN SI
SI R >= 5 ALORS // Traite le cas 50 et prépare 60, 70 et 80
    NbRomain = NbRomain + L
    R = R - 5
FIN SI
POUR 1 A R FAIRE // Traite les cas 10, 20, 30, 60, 70 et 80
    NbRomain = NbRomain + X
FIN POUR
UN :
R = NbSaisi // R contient le dernier chiffre du nombre saisi
SI R = 0 ALORS ALLER A TERMINE FIN SI
SI R = 9 ALORS
    NbRomain = NbRomain + IX
    ALLER A TERMINE
FIN SI
SI R = 4 ALORS
    NbRomain = NbRomain + IV
    ALLER A TERMINE
FIN SI
SI R >= 5 ALORS
    NbRomain = NbRomain + V
    R = R - 5
FIN SI
POUR 1 A R FAIRE
    NbRomain = NbRomain + I
FIN POUR
TERMINE :
AFFICHER « le chiffre romain correspondant au nombre saisi est « NbRomain

```

Voilà, si vous êtes arrivé jusque là et que vous avez tout compris, bravo vous êtes qualifié pour devenir programmeur. Au final notre algorithme au complet donne ceci :

## DONNEES

DECLARE NbSaisi COMME nombre entier

DECLARE NbRomain COMME chaîne de caractères

DECLARE R COMME nombre entier

## CODE

DEBUT :

AFFICHER « Saisissez un nombre compris entre 1 et 3999 »

LIRE NbSaisi

TANT QUE NbSaisi < 1 OU NbSaisi > 3999 FAIRE

    AFFICHER « vous devez saisir un nombre compris entre 1 et 3999 »

    LIRE NbSaisi

FIN TANT QUE

MILLE :

R = NbSaisi / 1000

// R contient le chiffre des milliers du nombre

//saisi

SI R=0 ALORS ALLER A CENT FIN SI

//Si NbSaisi<1000 on passe aux centaines

NbSaisi = NbSaisi – ( R x 1000 )

// On enlève le chiffre des milliers du nombre

//saisi

POUR 1 A R FAIRE

// Traite les cas 1000, 2000, et 3000

    NbRomain = NbRomain + M

FIN POUR

CENT :

R = NbSaisi / 100

// R contient le chiffre des centaines du nombre

//saisi

SI R=0 ALORS ALLER A DIX FIN SI

//Si NbSaisi<100 on passe aux dizaines

NbSaisi = NbSaisi – ( R x 100 )

// On enlève le chiffre des centaines du nombre

//saisi

SI R = 9 ALORS

// Traite le cas 900 et passe aux dizaines

    NbRomain = NbRomain + CM

    ALLER A DIX

FIN SI

SI R = 4 ALORS

// Traite le cas 400 et passe aux dizaines

    NbRomain = NbRomain + CD

    ALLER A DIX

FIN SI

SI R >= 5 ALORS

// Traite le cas 500 et prépare 600, 700 et 800

    NbRomain = NbRomain + D

    R = R – 5

FIN SI

POUR 1 A R FAIRE

// Traite les cas 100, 200, 300, 600, 700 et 800

    NbRomain = NbRomain + C

FIN POUR

DIX :

R = NbSaisi / 10

// R contient le chiffre des dizaines du nombre

//saisi

SI R = 0 ALORS ALLER A UN FIN SI

// Si NbSaisi < 10 on passe au chiffre des unités

NbSaisi = NbSaisi – ( R x 10 )

// On enlève le chiffre des dizaines de NbSaisi

SI R = 9 ALORS

// Traite le cas 90 et passe au chiffre des unités

    NbRomain = NbRomain + XC

    ALLER A UN

FIN SI

SI R = 4 ALORS

// Traite le cas 40 et passe au chiffre des unités

```

        NbRomain = NbRomain + XL
FIN SI
SI R >= 5 ALORS // Traite le cas 50 et prépare 60, 70 et 80
    NbRomain = NbRomain + L
    R = R - 5
FIN SI
POUR 1 A R FAIRE // Traite les cas 10, 20, 30, 60, 70 et 80
    NbRomain = NbRomain + X
FIN POUR
UN :
R = NbSaisi // R contient le dernier chiffre du nombre saisi
SI R = 0 ALORS ALLER A TERMINE FIN SI
SI R = 9 ALORS
    NbRomain = NbRomain + IX
    ALLER A TERMINE
FIN SI
SI R = 4 ALORS
    NbRomain = NbRomain + IV
    ALLER A TERMINE
FIN SI
SI R >= 5 ALORS
    NbRomain = NbRomain + V
    R = R - 5
FIN SI
POUR 1 A R FAIRE
    NbRomain = NbRomain + I
FIN POUR
TERMINE :
AFFICHER « le chiffre romain correspondant au nombre saisi est » NbRomain

```

Mais s'arrêter ici laisse comme un goût d'inachevé, vous ne trouvez pas ? Franchement, c'est du boulot de tâcheron, on fait quasiment 4 fois la même chose. Ne pourrait-on pas faire mieux, plus élégant ?

Si, on peut !

Mais pour cela, il va me falloir ajouter 3 nouvelles données : 1 nombre et 2 tableaux de caractères. Voici la déclaration des nouvelles variables.

## DONNEES

```

DECLARE NbSaisi COMME nombre entier
DECLARE NbRomain COMME chaîne de caractères
DECLARE R COMME nombre entier
DECLARE N COMME nombre entier
DECLARE Lettre1[4] COMME tableau de 4 caractères
DECLARE Lettre5[3] COMME tableau de 3 caractères

```

Le début du code ne change pas :

## CODE

```

DEBUT :
AFFICHER « Saisissez un nombre compris entre 1 et 3999 »
LIRE NbSaisi

```

TANT QUE NbSaisi < 1 OU NbSaisi > 3999 FAIRE  
 AFFICHER « vous devez saisir un nombre compris entre 1 et 3999 »  
 LIRE NbSaisi  
 FIN TANT QUE

Mais juste après, j'initialise mes nouvelles variables :

N = 3  
 Lettre1<sub>[0..3]</sub> = {I ; X ; C ; M} // Lettre1<sub>[0]</sub> = I ; Lettre1<sub>[1]</sub> = X, Lettre1<sub>[2]</sub> = C ...  
 Lettre5<sub>[0..2]</sub> = {V ; L ; D} /// Lettre5<sub>[0]</sub> = V ; Lettre5<sub>[1]</sub> = L, Lettre1<sub>[2]</sub> = D

Et je fais tout le traitement dans une seule boucle POUR. Lors de chacun des 4 passages successifs dans la boucle la valeur de N descendra de 3 à 0. N sera utilisé en même temps comme exposant de puissance de 10 pour les opérations mathématiques sur NbSaisi et comme indice dans les tableaux de caractères romains. Au premier passage dans la boucle le traitement des milliers, au deuxième les centaines etc.

POUR 1 A 4 FAIRE

```

R = NbSaisi / 10N
NbSaisi = NbSaisi - ( R x 10N )
SI R = 0 ALORS
  ALLER A SUIVANT
FIN SI
SI R = 9 ALORS
  NbRomain = NbRomain + Lettre1[N] + Lettre1[N+1]
  ALLER A SUIVANT
FIN SI
SI R = 4 ALORS
  NbRomain = NbRomain + Lettre1[N] + Lettre5[N]
  ALLER A SUIVANT
FIN SI
SI R >= 5 ET R < 9 ALORS
  NbRomain = NbRomain + Lettre5[N]
  R = R - 5
FIN SI
POUR 1 A R FAIRE
  NbRomain = NbRomain + Lettre1[N]
FIN POUR
SUIVANT :
N = N - 1

```

FIN POUR

Donc, le nouvel algo devient :

## DONNEES

DECLARE NbSaisi COMME nombre entier  
DECLARE NbRomain COMME chaîne de caractères  
DECLARE R COMME nombre entier  
DECLARE N COMME nombre entier  
DECLARE Lettre1<sub>[4]</sub> COMME tableau de 4 caractères  
DECLARE Lettre5<sub>[3]</sub> COMME tableau de 3 caractères

## CODE

DEBUT :

AFFICHER « Saisissez un nombre compris entre 1 et 3999 »

LIRE NbSaisi

TANT QUE NbSaisi < 1 OU NbSaisi > 3999 FAIRE

    AFFICHER « vous devez saisir un nombre compris entre 1 et 3999 »

    LIRE NbSaisi

FIN TANT QUE

N = 3

Lettre1<sub>[0..3]</sub> = {I ;X;C;M}

Lettre5<sub>[0..2]</sub> = {V;L;D}

POUR 1 A 4 FAIRE

    R = NbSaisi / 10<sup>N</sup>

    SI R = 0 ALORS ALLER A SUIVANT FIN SI // cette ligne n'est pas nécessaire

    NbSaisi = NbSaisi - ( R x 10<sup>N</sup> )

    SI R = 9 ALORS

        NbRomain = NbRomain + Lettre1<sub>[N]</sub> + Lettre1<sub>[N+1]</sub>

        ALLER A SUIVANT

    FIN SI

    SI R = 4 ALORS

        NbRomain = NbRomain + Lettre1<sub>[N]</sub> + Lettre5<sub>[N]</sub>

        ALLER A SUIVANT

    FIN SI

    SI R >= 5 ALORS

        NbRomain = NbRomain + Lettre5<sub>[N]</sub>

        R = R - 5

    FIN SI

    POUR 1 A R FAIRE

        NbRomain = NbRomain + Lettre1<sub>[N]</sub>

    FIN POUR

    SUIVANT :

    N = N - 1

FIN POUR

AFFICHER « le chiffre romain correspondant au nombre saisi est « NbRomain

C'est mieux, non ? Plus compact en tous cas. Mais on peut encore faire un peu mieux. On peut remplacer tous les branchements à SUIVANT par une opération différente mais qui assure le bon fonctionnement du programme. Et du coup, comme tous ces branchements ont disparu, l'étiquette SUIVANT : n'est plus nécessaire non plus.

Alors ? Vous avez deviné ? Par quoi peut-on remplacer tous les ALLER A SUIVANT tout en gardant un algorithme correct ? La réponse en page suivante ...



R = 0 , Bon sang, mais c'est bien sûr ! Et notre nouvel algo devient :

## CODE

DEBUT :

AFFICHER « Saisissez un nombre compris entre 1 et 3999 »

LIRE NbSaisi

TANT QUE NbSaisi < 1 OU NbSaisi > 3999 FAIRE

    AFFICHER « vous devez saisir un nombre compris entre 1 et 3999 »

    LIRE NbSaisi

FIN TANT QUE

N = 3

Lettre1<sub>[0..3]</sub> = {I ;X;C;M}

Lettre5<sub>[0..2]</sub> = {V;L;D}

POUR 1 A 4 FAIRE

    R = NbSaisi / 10<sup>N</sup>

    NbSaisi = NbSaisi – ( R x 10<sup>N</sup> )

    SI R = 9 ALORS

        NbRomain = NbRomain + Lettre1<sub>[N]</sub> + Lettre1<sub>[N+1]</sub>

        R = 0

    FIN SI

    SI R = 4 ALORS

        NbRomain = NbRomain + Lettre1<sub>[N]</sub> + Lettre5<sub>[N]</sub>

        R = 0

    FIN SI

    SI R > 4 ALORS

        NbRomain = NbRomain + Lettre5<sub>[N]</sub>

        R = R – 5

    FIN SI

    POUR 1 A R FAIRE

        NbRomain = NbRomain + Lettre1<sub>[N]</sub>

    FIN POUR

    N = N – 1

FIN POUR

AFFICHER « le chiffre romain correspondant au nombre saisi est « NbRomain

Là, je pense qu'on est à l'os.