

Thème numéro 12

Points essentiels traités dans le thème :

1. Algorithmique : utilisation d'un automate à états déterministe
2. Ecriture de code VBA
3. Fonctions *Asc*, *Ucase*, *Mid*, *Len*
4. Instructions *Select Case*, *For...Next*, *Exit For*
5. Utilisation des procédures de mise au point

La procédure présentée ici a pour but de convertir une référence de cellule de type *A1* en un numéro de colonne et un numéro de ligne, qui seront rangés dans deux variables entières distinctes.

Une telle procédure pourra servir à analyser une adresse de cellule utilisée dans un traitement complexe portant sur un tableau EXCEL. Surtout, elle permet d'introduire une technique algorithmique particulière, celle des *automates à états*

Remarque : le but essentiel de ce thème est de vous familiariser avec les instructions Select Case et For...Next.

Le principe des automates à état n'apparaîtra sans doute jamais dans un sujet d'examen. Si cet aspect de l'algorithmique ne vous passionne pas particulièrement, concentrez vous sur les nouveautés que vous apporte ce thème

- *Du point de vue du VBA : nouvelles instructions*
- *Du point de vue de l'interface utilisateur du debugger : utilisation d'espions*

12.1. Les automates à états déterministes

Ces automates permettent d'analyser des chaînes de caractères dont la structure est relativement simple.

Si la chaîne à analyser possède une structure complexe par exemple une structure de blocs, il est nécessaire de recourir à un *automate à pile*. La pile sert dans ce cas à mémoriser divers renseignements lors de l'ouverture d'un bloc à l'intérieur d'un autre bloc.

L'automate sur lequel nous allons travailler dans ce TD n'analyse que des structures simples, il est dit *déterministe* parce que les changements d'état sont entièrement déterminés par l'état courant et la catégorie du caractère traité.

(Un état sert à déterminer quelles opérations l'automate doit exécuter lorsqu'il rencontre certaines catégories de caractères dans la chaîne d'entrée)

Un automate à états déterministe est constitué :

1. d'un ensemble E d'états e_i constitué :

- d'un *état de départ* e_0
- d'un ensemble d'*états d'acceptation* E_a
- d'un ensemble d'*états d'erreur* E_r

2. d'un ensemble R de *règles*, chacune d'entre elles étant associée à un état et à une catégorie de caractères de la chaîne d'entrée

L'automate analyse une *chaîne d'entrée* de la manière suivante ;

- au début du travail, il se trouve dans l'état e_0
- Pour chaque caractère de la chaîne d'entrée, il applique la règle qui correspond à l'état dans lequel il se trouve et à la catégorie du caractère à traiter. Cette règle peut avoir l'un, l'autre ou les deux effets suivants :
 - modifier l'état de l'automate
 - exécuter une *action*

Ici, l'automate retenu possède cinq états :

- Etat 0 : attente de la première lettre de l'identification de colonne
- Etat 1 : Attente d'une autre lettre de l'identification de colonne ou du premier chiffre de l'identification de la ligne
- Etat 2 : Attente d'un nouveau chiffre de l'identification de la ligne
- Etat 3 : Une lettre a été rencontrée dans l'état 2
- Etat 4 : Le numéro de colonne calculé est trop grand
- Etat 5 : Le numéro de ligne calculé est trop grand.

L'unique état d'acceptation est l'état 2. Dans cet état, l'automate a déjà rencontré une identification de ligne et un numéro de colonne.

Tous les autres états sont des états d'erreur

12.2. Saisie du code VBA

- Lancez EXCEL sur un nouveau classeur.
- Créez dans ce classeur un *module*, comme vous l'avez fait lors du thème précédent:
 - Ouvrir l'éditeur Visual Basic, soit par <Alt><F11>, soit par **Développeur**→**Code**→**Visual Basic**
(**developer**→**Macro**→**Visual Basic**)
 - Faire un clic droit sur l'icône représentant le projet, puis activer **insertion**→**module** (**Insert**→**Module**)

Le texte de la procédure suit. Chacune de ses lignes est précédée d'un losange, pour les distinguer des lignes d'explications.

◆ Sub *cnvad*

⇒La procédure s'appellera *cnvad* (conversion d'adresse). L'éditeur ajoute automatiquement une paire de parenthèses pour délimiter les paramètres (inexistants) de la procédure, la ligne **End Sub**, et positionne le curseur d'insertion entre les deux lignes

```
◆ Dim etat As Integer
◆ Dim vncar As Integer
◆ Dim chainesaisie As String
◆ Dim i As Integer
◆ Dim nc As Integer
◆ Dim nl As Long
```

⇒ Ces instructions déclarent les variables utilisées par la procédure :

⇒ *etat* est l'état de l'automate

⇒ *vncar* est le code *ANSI* du caractère à traiter (valeur numérique du caractère)

⇒ *chainesaisie* est la chaîne de caractères à analyser

⇒ *i* est un indice qui permet de parcourir cette chaîne

⇒ *nc* est la variable entière dans laquelle sera rangé le numéro de colonne

⇒ *nl* est la variable entière dans laquelle sera rangé le numéro de ligne

```
◆ chainesaisie = InputBox( _
  "introduisez une référence de cellule")
```

⇒ Cette boîte de saisie simplifiée (pas de titre, pas de valeur par défaut) permet à l'opérateur de saisir la chaîne de caractères à analyser

```
◆ etat = 0
```

⇒ L'état est initialisé à la valeur correspondant à l'état de départ

```
◆ For i = 1 To Len(chainesaisie)
```

⇒ Cette instruction est le début de la boucle qui parcourt la chaîne à analyser

⇒ La fonction *Len* rend un entier, égal au nombre de caractères de son argument, qui doit être une chaîne de caractères

◆ `vncar = Asc (UCase _
(Mid(chainesaisie, i, 1)))`

⇒ La fonction **Mid** nécessite trois arguments :

- le premier est une chaîne de caractères **C**
- le second est un entier **P** qui représente une position dans la chaîne, le premier caractère de la chaîne a pour position **I**
- le troisième est un entier **L** qui représente une longueur

Cette fonction rend une chaîne de caractères extraite de la chaîne **C**, à partir de la position **P**, sur une longueur égale à **L**. Ici, elle rend une chaîne d'un seul caractère, qui est le **i^{ème}** caractère de la chaîne **chainesaisie**

⇒ La fonction **UCase** reçoit comme unique argument une chaîne, et rend une copie de cette chaîne, dans laquelle toutes les lettres minuscules ont été remplacées par des lettres majuscules.

⇒ La fonction **Asc** reçoit comme unique argument une chaîne, et rend un entier égal au code **ANSI** du premier caractère de cette chaîne.

⇒ Cette instruction range donc dans **vncar** le code **ANSI** du **i^{ème}** caractère de la chaîne **chainesaisie**, après une éventuelle transformation en majuscule.

◆ `Select Case vncar`

⇒ Cette instruction débute un bloc d'instructions **Select Case**, destinées à traiter les diverses catégories de caractères

◆ `Case Asc("A") To Asc("Z")`

⇒ Cette instruction **Case** débute la séquence d'instructions qui traite les caractères alphabétiques. (On reconnaît les caractères alphabétiques à ce que le contenu de **vncar**, qui est le code **ANSI** du caractère traité, éventuellement transformé en majuscule, est compris entre les codes **ANSI** des caractères **A** et **Z**)

Attention : Excel ne passera pas automatiquement en majuscule les constantes de chaînes de caractères telles que "A" ou "Z". Vous devez les introduire en majuscule vous-même

◆ `If etat = 0 Then etat = 1`

⇒ Si l'automate est dans l'état **0**, il attend la première lettre de la désignation de colonne. Celle-ci vient d'arriver ; l'automate doit passer à l'état **1** (attente des lettres suivantes ou du premier chiffre du numéro de ligne)

◆ `If etat = 2 Then`
 ◆ `etat = 3`
 ◆ `Exit For`
 ◆ `End If`

⇒ Si l'automate est dans l'état 2, il ne peut accepter qu'un chiffre. L'apparition d'une lettre est donc une erreur, qui est signalée par le passage de l'automate à l'état 3.

⇒ L'instruction ***Exit For*** provoque une sortie immédiate de la boucle ***For***.

◆ `nc = nc * 26 + vncar - _`
`Asc("A") + 1`

⇒ Ici l'automate est nécessairement dans l'état 1 (le passage dans l'un quelconque des autres états : 2, 3, 4 et 5, aurait provoqué la fin de la boucle ***For***). L'action qui correspond à la réception d'une lettre dans cet état consiste à :

- multiplier par 26 l'ancienne valeur de ***nc***
- ajouter à la valeur ainsi obtenue un nombre compris entre 1 et 26, qui correspond à la lettre

Pour fixer les idées, examinons par exemple l'analyse de la chaîne "***BC***" :

- A l'origine, ***nc*** est nul
- Après réception du caractère ***B***, cette valeur nulle est multipliée par 26, ce qui ne la change pas, puis la valeur 2, qui correspond à la lettre ***B***, lui est ajoutée, ce qui donne la valeur 2
- Après réception du caractère ***C***, la valeur 2 est multipliée par 26, ce qui donne 52, puis la valeur 3, qui correspond à la lettre ***C***, lui est ajoutée. On obtient ainsi 55

On vérifiera sans peine que la colonne ***BC*** est bien la 55^{ième} colonne d'un tableau EXCEL.

Explication de l'obtention d'un nombre compris entre 1 et 26 correspondant à la lettre

Caractère	vncar	vncar – Asc("A")	vncar – Asc("A")+1
A	65	0	1
B	66	1	2
C	67	2	3
D	68	3	4
E	69	4	5
F	70	5	6
G	71	6	7
H	72	7	8
I	73	8	9
J	74	9	10
K	75	10	11
L	76	11	12
M	77	12	13
N	78	13	14
O	79	14	15
P	80	15	16
Q	81	16	17
R	82	17	18
S	83	18	19
T	84	19	20
U	85	20	21
V	86	21	22
W	87	22	23
X	88	23	24
Y	89	24	25
Z	90	25	26

```

♦           If nc > 16384 Then
♦               etat = 4
♦               Exit For
♦           End If

```

⇒ Depuis EXCEL 2007, un numéro de colonne ne peut dépasser 16384. Si *nc* vient à dépasser 16384, il s'agit d'une erreur. L'automate le signale en passant à l'état 4.

⇒ L'instruction *Exit For* provoque une sortie immédiate de la boucle *For*.

```

♦           Case Asc("0") To Asc("9")

```

⇒ Cette instruction *Case* débute la séquence d'instructions qui traite les caractères numériques (On reconnaît les caractères numériques à ce que le contenu de *vncar*, qui est le code *ANSI* du caractère à traiter, éventuellement transformé en majuscule, est compris entre les codes *ANSI* des caractères *0* et *9*)

```

♦           If etat = 0 Then Exit For
♦           If etat = 1 Then etat = 2

```

⇒ L'automate ne peut se trouver que dans l'état 0, l'état 1 ou l'état 2 (le passage à l'un quelconque des autres états : 3, 4 ou 5, aurait provoqué la fin de la boucle d'analyse). Il faut désormais qu'il passe à l'état 2, car un chiffre a été rencontré.

- S'il est dans l'état 0, il attend une lettre. L'automate sort de la boucle **For** dans l'état **0**, qui est un état d'erreur
- S'il est dans l'état **1**, il doit passer à l'état **2**, car il a rencontré un chiffre

```

♦           nl = nl * 10 + vncar - _
               Asc("0")

```

⇒ L'action qui correspond à la réception d'un chiffre dans l'état 2 consiste à :

- multiplier par 10 l'ancienne valeur de *nl*,
- ajouter à la valeur obtenue un nombre compris entre 0 et 9, qui correspond au chiffre.
- Examinons l'analyse de la chaîne “ **123** ” :
- A l'origine, *nl* vaut zéro

- A la réception du chiffre **1**, cette valeur zéro est multipliée par 10, ce qui ne la change pas, puis la valeur 1, qui correspond au chiffre **1**, lui est ajoutée. On obtient ainsi l'entier 1.
- A la réception du chiffre **2**, la valeur 1 est multipliée par 10, ce qui donne 10, puis la valeur 2, qui correspond au chiffre **2**, lui est ajoutée. On obtient ainsi l'entier 12.
- A la réception du chiffre **3**, la valeur 12 est multipliée par 10, ce qui donne 120, puis la valeur 3, qui correspond au chiffre **3**, lui est ajoutée. On obtient ainsi l'entier 123.

Explication de l'obtention d'un entier compris entre 0 et 9, qui correspond au chiffre

Caractère	vncar	vncar – Asc("0")
0	48	0
1	49	1
2	50	2
3	51	3
4	52	4
5	53	5
6	54	6
7	55	7
8	56	8
9	57	9

```

◆          If nl > 1048576 Then
◆              etat = 5
◆              Exit For
◆          End If

```

⇒ Depuis EXCEL2007, un numéro de ligne ne peut dépasser 1048576. Si *nl* vient à dépasser 1048576, il s'agit d'une erreur. L'automate le signale en passant à l'état 5

⇒ L'instruction **Exit For** provoque une sortie immédiate de la boucle **For**.

◆ `End Select`

⇒ Cette instruction termine le bloc d'instructions ouvert par l'instruction *Select Case vncar*.

◆ `Next i`

⇒ Cette instruction termine la boucle *For*, et fait progresser *i*. Si *i* n'est pas devenu supérieur au nombre de caractères de la chaîne à analyser, un nouveau tour de boucle analysera le caractère suivant. Dans le cas contraire, l'analyse est terminée, et on peut passer à l'affichage des résultats.

◆ `Select Case etat`

⇒ En fonction de la valeur de la variable *etat* en sortie de boucle, on affiche, soit un message d'erreur, soit le résultat

◆ `Case 0`

◆ `MsgBox chainesaisie + _`

◆ `" : pas de désignation de colonne"`

⇒ Si on sort de la boucle dans l'état **0**, l'automate n'a pas rencontré de lettre en début de chaîne. C'est une erreur. On l'annonce.

Les espaces au début des textes ont pour but d'éviter que ceux-ci soient collés au contenu de la variable **chainesaisie** lors de l'affichage, et de rendre celui-ci plus lisible.

◆ `Case 1`

◆ `MsgBox chainesaisie + _`

◆ `" : pas de numéro de ligne"`

⇒ Si on sort de la boucle dans l'état **1**, aucun chiffre n'a été rencontré. C'est une erreur. On l'annonce.

◆ `Case 2`

◆ `MsgBox chainesaisie + ___`

◆ `" : colonne " + Str (nc) + _`

◆ `" : ligne " + Str (nl)`

⇒ Si on est sorti de la boucle dans l'état **2**, les deux identifications étant présentes, le résultat est a priori correct. On l'affiche. Attention : il faut des espaces dans les textes autour de toutes les expressions affichées.

◆ Case 3

◆

```
MsgBox chainesaisie + _  
" : lettre dans le numéro de ligne"
```

⇒ Si on est sorti de la boucle dans l'état 3, on a rencontré une lettre alors qu'on attendait un chiffre. C'est une erreur. On l'annonce.

◆ Case 4

◆

```
MsgBox chainesaisie + _  
" : numéro de colonne trop grand"
```

⇒ Si on est sorti de la boucle dans l'état 4, c'est que l'opérateur a introduit un identificateur de colonne erroné, qui a conduit à un numéro de colonne trop grand. C'est une erreur. On l'annonce.

◆ Case 5

◆

```
MsgBox chainesaisie + _  
" : numero de ligne trop grand"
```

⇒ Si on est sorti de la boucle dans l'état 5, c'est que l'opérateur a introduit un identificateur de ligne erroné, qui a conduit à un numéro de ligne trop grand. C'est une erreur. On l'annonce.

◆ End Select

⇒ L'affichage des résultats est terminé. Cette instruction ferme le bloc ouvert par l'instruction *Select Case etat*.

◆ End Sub

⇒ La procédure est terminée.

Remarque : cette ligne a déjà été introduite automatiquement par l'Editeur VBA.

Il vous reste à présent à sauvegarder votre classeur et à vérifier en pas-à-pas le bon fonctionnement de votre procédure. Les variables à espionner sont toutes les variables déclarées dans l'en-tête de la procédure. Il pourra également être intéressant de suivre l'évolution de l'expression **Mid(chainesaisie , i , 1)** qui affiche le caractère courant.

Un *espion* porte sur la valeur d'une *expression*. Cette expression doit avoir pour type l'un des types scalaires prédéfinis : nombre, chaîne ou booléen. Bien entendu, l'expression peut être réduite à une simple variable.

Pour ajouter un espion portant sur la valeur d'une expression qui existe déjà dans votre code :

- Ouvrez la fenêtre **espions**, en activant la rubrique de menu : **Affichage**→**Fenêtre espions** (**View**→**Watch Window**)
- Sélectionnez dans votre code l'expression à espionner
- Faites un clic droit dans la fenêtre « espions », pour ouvrir le menu contextuel de cette fenêtre.
- Activez la rubrique **Ajouter un espion** (**Add Watch**)

Une boîte de dialogue intitulée *Ajouter un espion* s'ouvre. L'expression sélectionnée apparaît dans la boîte de saisie **Expression** de cette boîte de dialogue. Cliquez sur le bouton **Ok**

Pour ajouter un espion portant sur une expression qui ne figure pas dans votre code, vous devriez introduire vous-même cette expression dans la boîte de saisie **Expression** de la boîte de dialogue.

La zone **Contexte** de la boîte de dialogue permet de préciser le module et la procédure ou fonction dans lesquels se trouve l'expression. Par défaut, il s'agit du module affiché à l'écran et de la procédure ou fonction dans laquelle se trouve le curseur. Vous n'avez donc pas à modifier cette zone.

La zone **Type d'espion** (**Watch type**) de la boîte de dialogue permet de préciser l'action à réaliser. On pourrait ainsi introduire un point d'arrêt conditionnel lorsqu'une expression espionnée, qui doit alors être booléenne prend la valeur **True**, ou lorsque la valeur de l'expression espionnée change. L'action sélectionnée par défaut nous satisfait. Elle consiste à afficher simplement la valeur de l'expression espionnée.

Cette procédure ne contient pas de boucle imposant à l'opérateur d'introduire une valeur correcte. Il faudra donc la relancer après chaque essai. Le fait de commencer les essais par les cas d'erreurs aura donc moins d'importance que dans le cas du thème numéro 11.

Annexe : texte du module Module1

```

Sub cnvad()
`actualisation pour Excel 2019
`où une feuille Excel contient 16384 colonnes et 1048576 lignes
  Dim etat As Integer, vncar As Integer
  Dim chainesaisie As String
  Dim i As Integer, nc As Integer
  Dim nl As Long
  chainesaisie = InputBox( _
    "introduisez une référence de cellule")
  etat = 0
  For i = 1 To Len(chainesaisie)
    vncar = Asc(UCase _
      (Mid(chainesaisie, i, 1)))
    Select Case vncar
      Case Asc("A") To Asc("Z")
        If etat = 0 Then etat = 1
        If etat = 2 Then
          etat = 3
          Exit For
        End If
        nc = nc * 26 + vncar - _
          Asc("A") + 1
        If nc > 16384 Then
          etat = 4
          Exit For
        End If
      Case Asc("0") To Asc("9")
        If etat = 0 Then Exit For
        If etat = 1 Then etat = 2
        nl = nl * 10 + vncar - _
          Asc("0")
        If nl > 1048576 Then
          etat = 5
          Exit For
        End If
    End Select
  Next i
  Select Case etat
    Case 0
      MsgBox chainesaisie + _
        " : pas de désignation de colonne"
    Case 1
      MsgBox chainesaisie + _
        " : pas de numéro de ligne"
    Case 2
      MsgBox chainesaisie + _
        " : colonne " + Str(nc) + _
        " ligne : " + Str(nl)
    Case 3
      MsgBox chainesaisie + _
        " : lettre dans le numéro de ligne"
    Case 4
      MsgBox chainesaisie + _
        " : numéro de colonne trop grand"
    Case 5
      MsgBox chainesaisie + _
        " : numéro de ligne trop grand"
  End Select
End Sub

```