9/8.4.1

Listage des variables d'un programme

Dans ce paragraphe, vous trouverez un logiciel fort utile au développement des programmes en Basic. Ce logiciel établit la liste des variables d'un programme et les lignes où ces variables sont modifiées.

COMMENT UTILISER LE PROGRAMME ?

Le programme est écrit en Basic. Les programmes à analyser doivent au préalable être stockés sur disque sous forme ASCII en spécifiant le format « ,A » dans la commande SAVE. Par exemple, la commande :

SAVE "MONPROG,A"

sauvegarde le programme qui se trouve en mémoire dans le fichier MONPROG sous forme ASCII.

Entrez le listing suivant :

```
1000 REM ------------------------------------------
1010 REM Reference des variables d'un programme BASIC
1020 REM ------------------------------------------
1030 REM
1040 DIM t1$(500),t2(500)
1050 MODE 2
1060 PRINT"Reference des variables d'un programme"
1070 PRINT"--------------------------------------"
1080 PRINT:PRINT
1090 INPUT "Nom du fichier ASCII a analyser : ";n$
1100 PRINT:INPUT"Sortie sur 1)Ecran ou 2)Imprimante : ";rep
1110 OPENIN n$
1120 REM
1130 z=0
1140 LOCATE 1,9
1150 PRINT"Nombre de lignes analysees : "
1160 WHILE NOT EOF
```

```
1170    z=z+1
1180    LOCATE 29,9
1190    PRINT z
1200    LINE INPUT #9,l$
1210    p=0
1220    p=INSTR(p+1,l$,"=")
1230    IF p<>0 THEN GOSUB 1840 'Extraction
1240    IF p<>0 THEN 1220 'Poursuite extraction
1250 WEND
1260 CLOSEIN
1270 REM
1280 REM -----------
1290 REM Classement
1300 REM -----------
1310 REM
1320 PRINT:PRINT"Classement des donnees en cours ..."
1330 bis=1
1340 WHILE bis=1
1350    bis=0
1360    FOR j=1 TO i-1
1370      IF t1$(j)<=t1$(j+1) THEN 1410
1380      bis=1
1390      a$=t1$(j):t1$(j)=t1$(j+1):t1$(j+1)=a$
1400      a=t2(j):t2(j)=t2(j+1):t2(j+1)=a
1410    NEXT j
1420 WEND
1430 REM
1440 REM ------------------------------------
1450 REM Affichage/Impression du resultat
1460 REM ------------------------------------
1470 REM
1480 IF rep=2 THEN 1660 'Impression
1490 CLS
1500 PRINT"Les variables du fichier "n$" sont les suivantes
:"
1510 PRINT:PRINT"Variable        Ligne(s)"
1520 PRINT"---------       ---------------------------------
--------"
1530 FOR j=1 TO i
1540    PRINT t1$(j);
1550    PRINT SPACE$(15-LEN(t1$(j)));
1560    PRINT t2(j);
1570    k=1
1580    WHILE t1$(j)=t1$(j+k)
1590      PRINT "/";t2(j+k);
1600      k=k+1
1610    WEND
1620    PRINT
1630    IF k<>1 THEN j=j+k-1
1640 NEXT j
```

```
1650 GOTO 1820
1660 CLS:PRINT"Impression en cours ..."
1670 PRINT#8,"Les variables du fichier "n$" sont les suivant
es :"
1680 PRINT#8:PRINT#8,"Variable        Ligne(s)"
1690 PRINT#8,"---------        ------------------------------
----------"
1700 FOR j=1 TO i
1710   PRINT #8,t1$(j);
1720   PRINT #8,SPACE$(15-LEN(t1$(j)));
1730   PRINT #8,t2(j);
1740   k=1
1750   WHILE t1$(j)=t1$(j+k)
1760     PRINT #8,"/";t2(j+k);
1770     k=k+1
1780   WEND
1790   PRINT#8
1800   IF k<>1 THEN j=j+k-1
1810 NEXT j
1820 END
1830 REM
1840 REM
1850 REM ------------------------
1860 REM Extraction des variables
1870 REM ------------------------
1880 REM
1890 sp=p:bad=0
1900 ch$=MID$(l$,sp,1)
1910 WHILE (ch$<>" ") AND (ch$<>":")
1920   ch$=MID$(l$,sp,1)
1930   IF (ASC(ch$)=34) OR (ch$=">") OR (ch$="<") THEN bad=1

1940   IF (ASC(ch$)<91) AND (ASC(ch$)>64) THEN bad=1
1950   sp=sp-1
1960 WEND
1970 IF bad=1 THEN 2080
1980 i=i+1
1990 t1$(i)=MID$(l$,sp+2,p-sp-2)
2000 WHILE LEFT$(t1$(i),1)="("
2010   t1$(i)=RIGHT$(t1$(i),LEN(t1$(i))-1)
2020 WEND
2030 sp=1
2040 WHILE MID$(l$,sp,1) <> " "
2050   sp=sp+1
2060 WEND
2070 t2(i)=VAL(MID$(l$,1,sp-1))
2080 RETURN
```

Sauvegardez-le sur disque sous forme ASCII en tapant :

SAVE « XREF »,A

A titre de test, nous vous suggérons d'exécuter le programme sur lui-même.

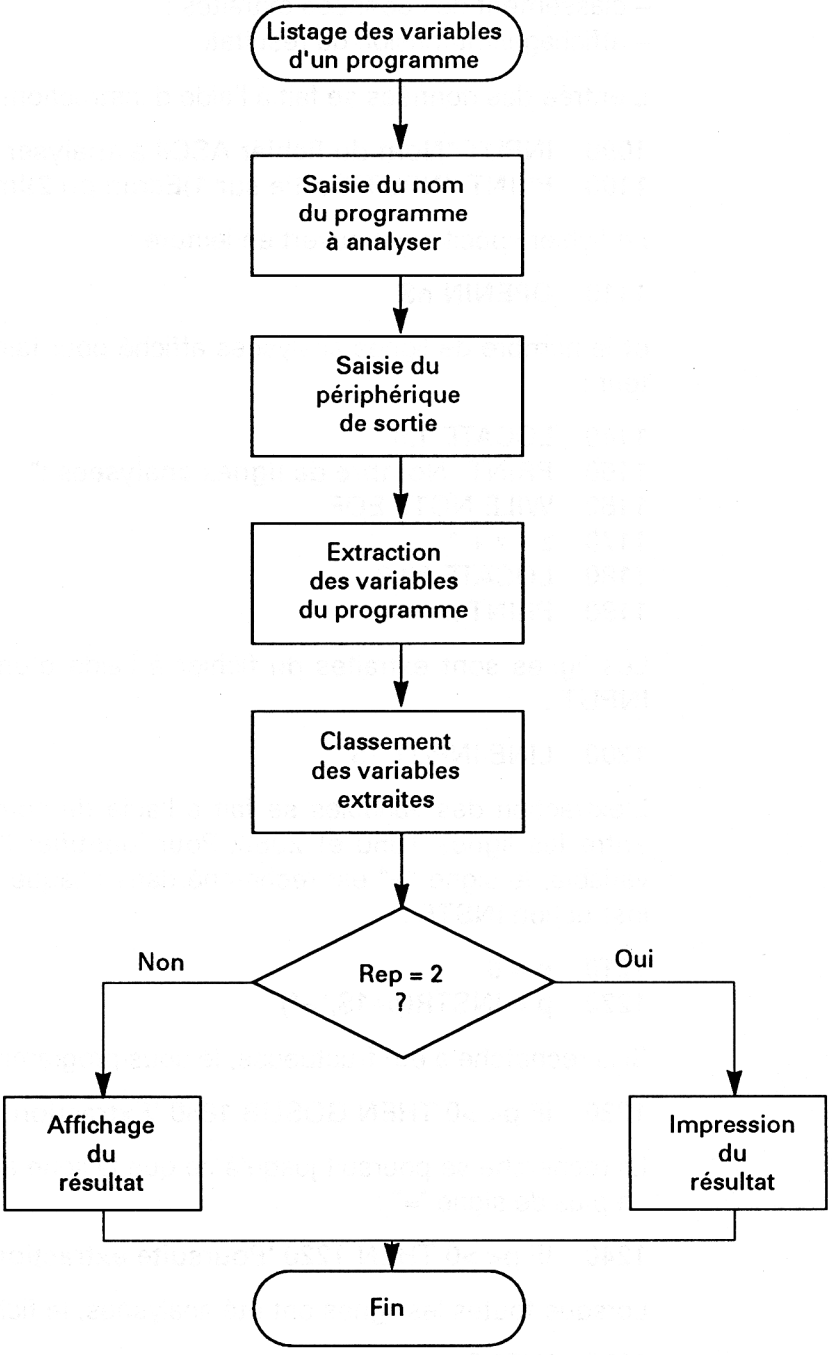
Lancez le programme et répondez XREF à la question « Nom du fichier ASCII à analyser ». Choisissez une sortie écran ou imprimante. Si vous avez saisi correctement le programme, le résultat sera le suivant :

```
Les variables du fichier xref sont les suivantes :

Variable        Ligne(s)
---------       ----------------------------------------
a               1400
a$              1390
bad             1890 / 1930 / 1940 / 1970
bis             1330 / 1340 / 1350 / 1380
ch$             1900 / 1920 / 1930 / 1930
i               1980
j               1360 / 1530 / 1630 / 1700 / 1800
k               1570 / 1600 / 1740 / 1770
p               1210 / 1220
rep             1480
sp              1890 / 1950 / 2030 / 2050
t1$(i)          1990 / 2010
t1$(j)          1390 / 1580 / 1750
t1$(j+1)        1390
t2(i)           2070
t2(j)           1400
t2(j+1)         1400
z               1130 / 1170
```

LE PROGRAMME EN DÉTAIL

La logique du programme apparaît dans l'ordinogramme suivant :



Le programme se divise en quatre parties :

– entrée des données (nom du fichier à analyser et périphérique de sortie) ;
– extraction des variables ;
– classement des données extraites ;
– affichage/impression du résultat.

L'entrée des données se fait à l'aide d'instructions PRINT/INPUT :

```
1090  INPUT "Nom du fichier ASCII à analyser : ";n$
1100  PRINT :INPUT"Sortie sur 1)Ecran ou 2)Imprimante : ";rep
```

Le fichier spécifié est ouvert en lecture :

```
1110  OPENIN n$
```

et le nombre de lignes analysées affiché pour faire patienter l'utilisateur :

```
1140  LOCATE 1,9
1150  PRINT "Nombre de lignes analysees :"
1160  WILE NOTE EOF
1170  z = z + 1
1180  LOCATE 29,9
1190  PRINT z
```

Les lignes sont extraites du fichier à l'aide d'une instruction LINE INPUT :

```
1200  LINE INPUT 1$
```

L'extraction des variables se fait à l'aide du sous-programme situé entre les lignes 1850 et 2080. Pour identifier l'initialisation d'une variable, le signe "=" est recherché dans chaque ligne à l'aide d'une instruction INSTR :

```
1210  p = 0
1220  p = INSTR(p+1$,"=")
```

Si la recherche a été fructueuse, le sous-programme est activé :

```
1230  IF p< >0 THEN GOSUB 1840 'Extraction
```

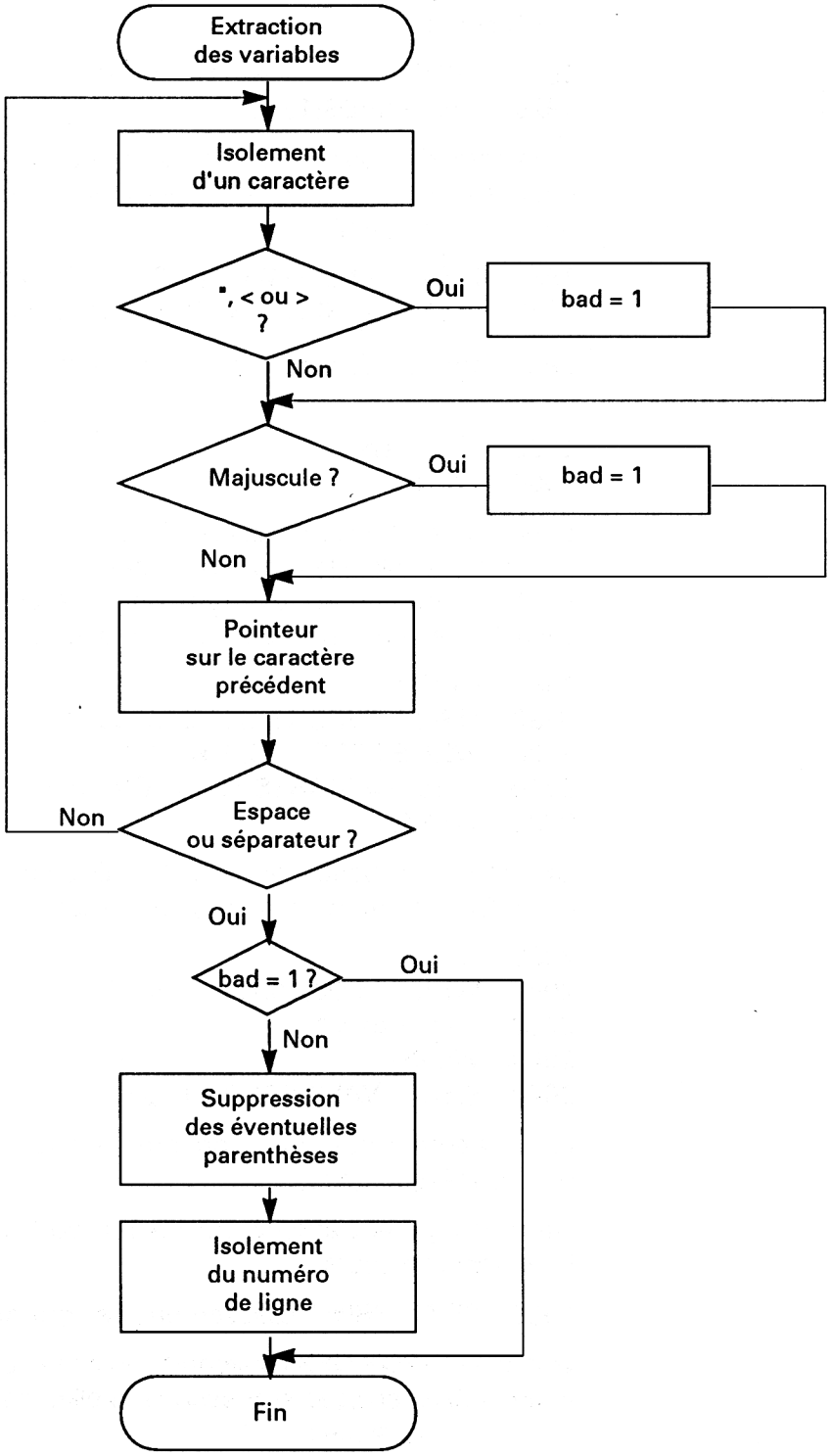
La recherche se poursuit jusqu'à ce que la ligne analysée ne contienne plus de signe "=" :

```
1240  IF p< >0 THEN 1220 'Poursuite extraction
```

Lorsque toutes les lignes ont été analysées, le fichier est fermé :

```
1250  WEND
1260  CLOSEIN
```

La logique du programme d'extraction apparaît dans l'ordinogramme suivant :



La première boucle **WHILE WEND** permet de trouver le début du nom de la variable pointée par **sp**. Dans cette boucle, **sp** est décrémenté jusqu'à ce que le caractère pointé soit un blanc ou un caractère séparateur :

```
1910   WILE (ch< >" ") AND (ch$< >":")
1920     ch$=MID$(1$,sp,1)
```

Si le caractère pointé est un guillemet, un signe supérieur ou inférieur, la variable **bad** est initialisée à un. De même si le caractère est une lettre majuscule :

```
1930   IF (ASC(ch$)=34) OR(ch$=">") OR (ch$="<") THEN bad = 1
1940   IF (ASC(ch$)<91) AND (ASC(ch$)>64) THEN bad = 1
```

Le pointeur de caractère est décrémenté et la boucle de recherche prend fin :

```
1950     sp = sp-1
1960   WEND
```

Si en fin de boucle, la variable **bad** vaut 1, l'expression isolée n'est pas une variable. Dans le cas contraire, une seconde boucle **WHILE WEND** supprime les éventuels caractères "(" situés en début de variable. Le nom de la variable est stocké dans **t1$** :

```
1980   i = i +1
1990   t1$(i) = MID$(1$,sp + 2,p - sp - 2)
2000   WHILE LEFT$(t1$(i),1) = "("
2010     t1$(i) = RIGHT$(t1$(i),LEN(t1$(i)) - 1)
2020   WEND
```

Une troisième boucle **WHILE WEND** permet d'isoler et de stocker dans **t2$** le numéro de la ligne concernée :

```
2030   sp = 1
2040   WHILE MID$(1$,sp,1) < > " "
2050     sp = sp + 1
2060   WEND
2070   t2$(i) = VAL(MID$(1$,1,sp - 1))
```

La troisième partie du programme est responsable du classement des données extraites.

Un message est affiché sur l'écran pour informer l'utilisateur de l'opération en cours :

```
1320   PRINT:PRINT"Classement des données en cours..."
```

Une boucle **FOR NEXT** parcourt toutes les entrées du tableau **t1$** et inverse deux entrées successives si elles ne sont pas classées dans l'ordre alphabétique :

```
1360  FOR j = 1 to i - 1
1370    IF t1$(j)< = t1$(j + 1) THEN 1410
1380    bis = 1
1390    a$ = t1$(j):t1$(j) = t1$(j + 1):t1$(j + 1) = a$
1400    a = t2(j):t2(j) = t2(j + 1):t2(j + 1) = a
1410  NEXT j
```

Le boucle **FOR NEXT** est imbriquée dans une boucle **WHILE WEND** qui prend fin lorsque le parcours complet du tableau **t1$** n'a donné lieu à aucune inversion :

```
1340  WHILE bis = 1
1350    bis = 0
        ...
1420  WEND
```

La dernière partie du programme affiche sur l'écran, ou imprime (selon la valeur de la variable **rep**) les données extraites et classées.

Si l'utilisateur a demandé une impression, le programme se débranche en 1660 :

```
1480  IF rep = 2 THEN 1660 'Impression
```

Dans le cas contraire, le programme affiche l'en-tête des données :

```
1500  PRINT "Les variables du fichier "n$" sont les suivantes :"
1510  PRINT:PRINT"Variable                Ligne(s)"
1520  PRINT "--------- ---------------------------------"
```

Une boucle **FOR NEXT** parcourt et affiche les entrées du tableau **t1$** :

```
1530  FOR j = 1 TO i
1540    PRINT t1$(j) ;
1550    PRINT SPACE$(15 - LEN(t1$(j))) ;
```

La première ligne contenant la variable **t1$(j)** est affichée :

```
1560    PRINT t2(j) ;
```

Si une ou plusieurs autres lignes modifient la variable **t1$(j)**, cette (ces) ligne(s) est (sont) affichées(s) à l'aide d'une boucle **WHILE WEND** :

```
1580    WHILE t1$(j) = t1$(j + k)
1590      PRINT "/";t2(j + k) ;
1600      k = k + 1
1610      WEND
1620      PRINT
```

Dans ce cas, la variable j est initialisée pour pointer sur la prochaine variable :

```
1630        IF k< >1 THEN j =j + k - 1
```

La partie du programme responsable de l'impression est en tout point similaire à la partie affichage dont nous venons de parler, le périphérique de sortie étant l'imprimante au lieu de l'écran (#8). Nous n'y reviendrons pas.