

## 9/8.17

# Utilitaires de modification de programmes Basic majuscule et remplacer

---

Nous vous proposons, dans ce chapitre, de résoudre un problème qui apparaît de temps en temps au programmeur en langage Basic : rechercher la (ou les) ligne(s) où se trouve une variable, pour la remplacer par un autre nom, soit plus explicite, pour étudier le programme, soit beaucoup plus simple, pour gagner en rapidité. Par exemple, remplacer la variable machintruc, réelle, par m%, variable entière, donc plus rapide à traiter, et ce dans tout le programme.

Manuellement, ce type de travail est fastidieux, pour peu que le nombre de variables à remplacer soit grand.

Il est possible d'utiliser un traitement de texte performant, qui permette des recherches et remplacements dans tout le texte. A supposer que vous possédiez un tel traitement de texte, il sera nécessaire de sauvegarder le programme Basic à modifier en ASCII, par la commande :

```
SAVE "PROGRAM.ASC",A
```

de lancer le traitement de texte, de charger le programme précédemment sauvegardé, et d'effectuer la recherche, et le remplacement éventuel. Espérons que votre logiciel vous demande confirmation, sinon le remplacement de X par X% produira des remplacements du genre : NEXT deviendra NEX%T, créant d'innombrables SYNTAX ERROR dans tout le programme.

Aussi, la définition du terme informatique étant le traitement automatique de l'information, nous avons pensé à ceux qui, avant tout, ne possèdent pas de traitement de texte et seraient obligés de modifier manuellement toutes les variables, ou trouvent trop lourde l'utilisation d'un traitement de texte pour un tel travail. Rappelons avant tout que ce genre d'opérations sur des fichiers ASCII nécessite l'emploi d'un lecteur de disquette pour effectuer des accès en lecture sur le fichier et des accès en lecture sur le fichier transformé.

Le programme que nous vous proposons aura pour avantage d'être simple d'emploi, mais pêchera, nous l'admettons, par sa vitesse, puisque écrit en Basic. Vous pourrez tout de même l'accélérer en modifiant le nom

des variables utilisées, et en otant toutes les remarques qui ne sont là que pour vous permettre d'étudier le programme, et par là-même améliorer vos performances en programmation sur AMSTRAD-CPC.

Avant toute chose, un petit problème apparaît : pour une même variable, mais écrite tantôt en minuscules, tantôt en majuscules, il fallait utiliser deux fois le programme, une fois pour les variables en majuscules, modifiées en leur nouveau nom, puis pour les variables en minuscules, pour le même nom de remplacement. Aussi nous avons créé un petit programme qui permet de transformer toutes les lettres des variables d'un programme en majuscules.

## I. Le majusculeur

### L'ALGORITHME

Nous avons, dans un premier temps, pensé à prendre une à une les lettres de chacune des lignes Basic composant le programme, sachant que les instructions Basic sont automatiquement mises en majuscules par l'interpréteur Basic. Or, il s'avérait que les textes prévus pour l'affichage se trouvaient aussi en majuscules. Le mal était moindre, car ne posant pas de problème de fonctionnement au programme, mais gênant quand les minuscules sont nécessaires pour faciliter la lecture des textes à l'écran.

Il faut donc penser à repérer tout guillemet qui signale à l'interpréteur le début d'une chaîne de texte, ainsi que le guillemet de fin, mais parmi ces deux fonctions du caractère guillemet, il faut reconnaître celle de début et celle de fin.

Tout bon programmeur essaie toujours de placer un guillemet à la fin d'une chaîne de caractères. Aussi, le premier guillemet rencontré est le guillemet de début de chaîne, le suivant, le guillemet de fin de chaîne, le suivant, le début de la chaîne suivante, etc.

Ainsi, nous avons décidé de placer une variable drapeau qui permet de signaler où l'on se trouve dans la ligne Basic (champ instruction ou chaîne) et d'autoriser ou non la conversion en majuscules, ce qui donne la portion d'algorithme suivant pour une ligne Basic :

— DEBUT : TRANSFORMATION D'UNE LIGNE BASIC

— Positionner un indicateur à 0

CO

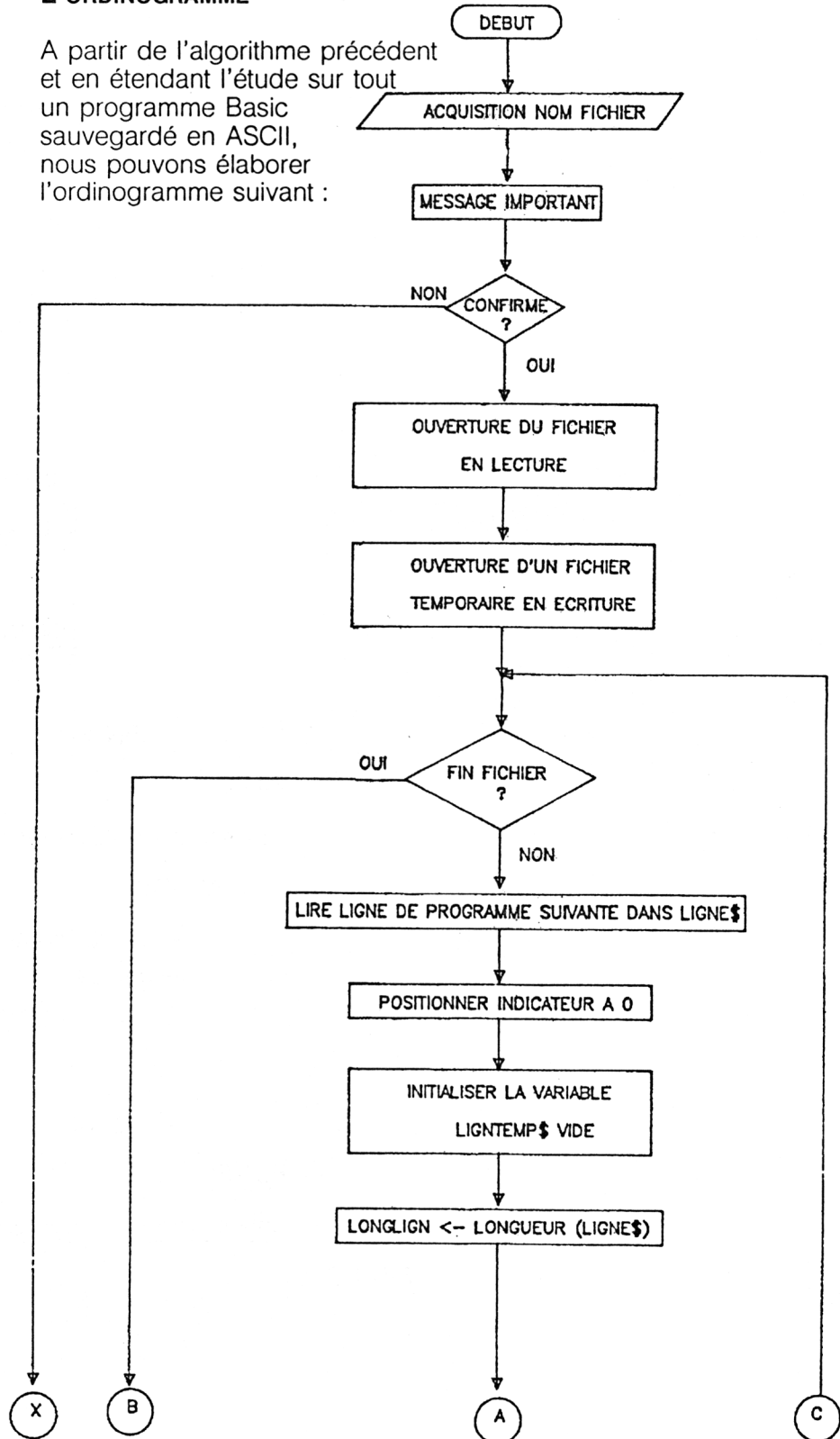
Afin de signaler une chaîne

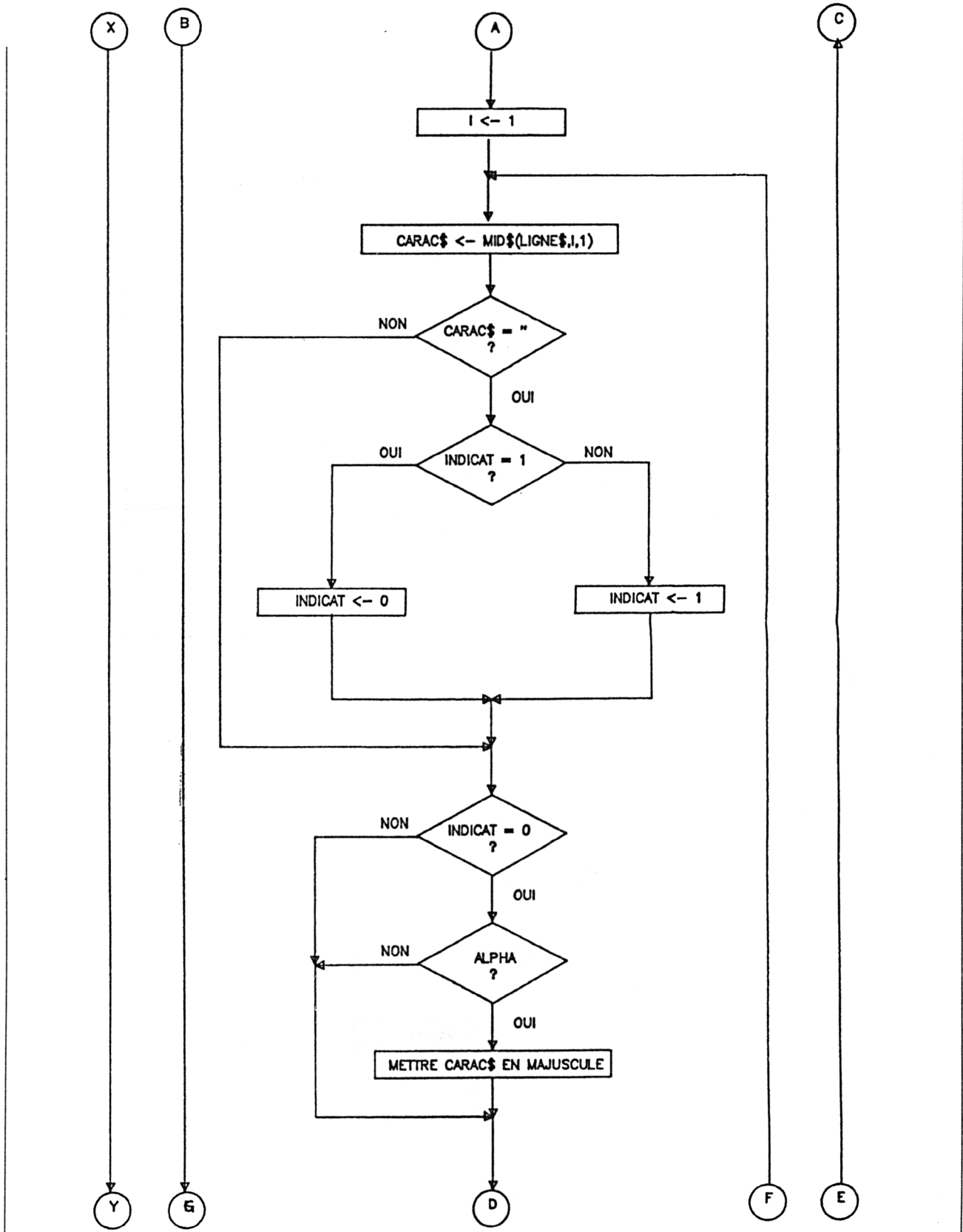
FIN CO

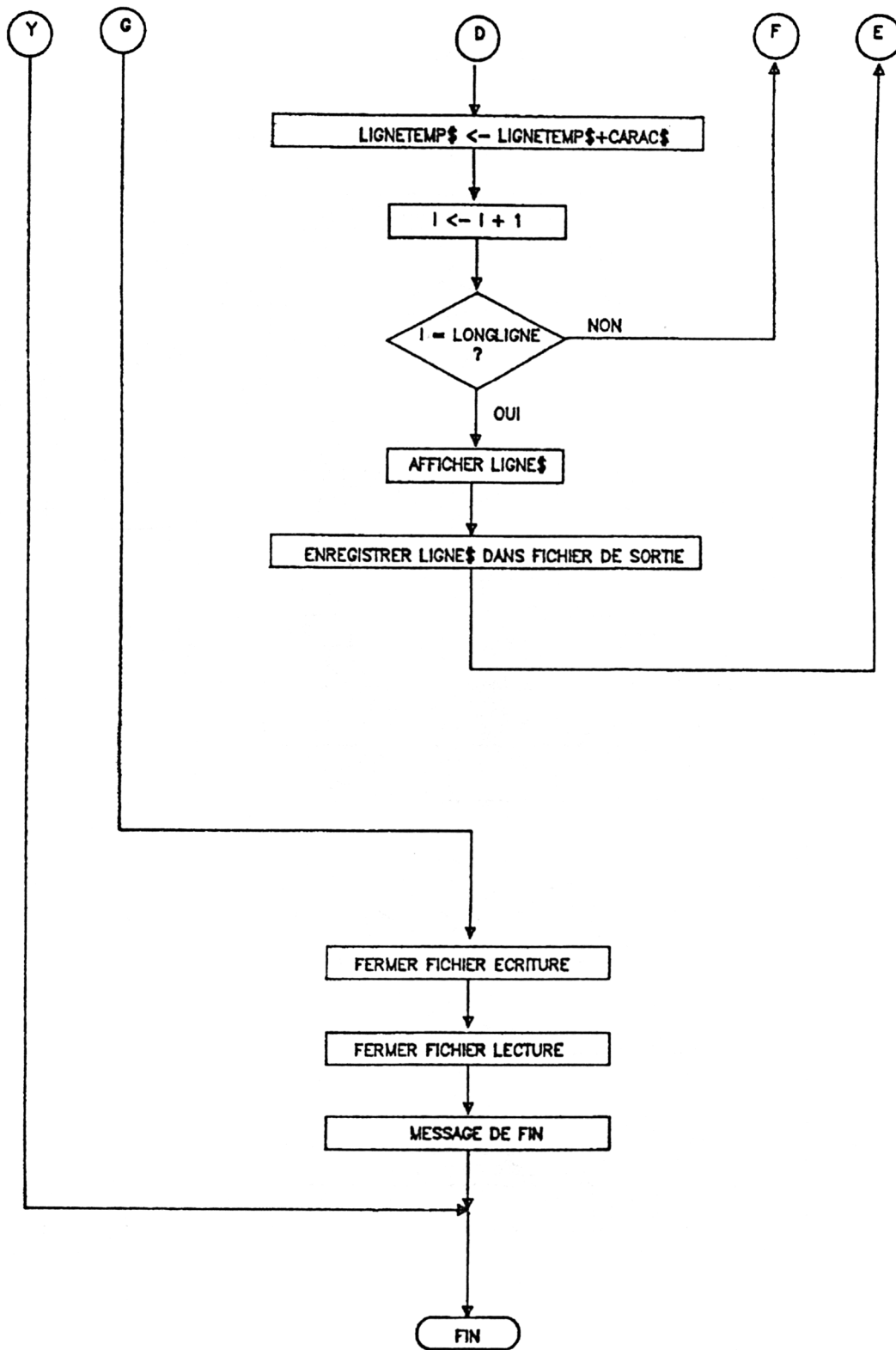
- Initialiser une variable caractère vide
  - CO
  - Afin de sauvegarder les caractères transformés
  - FIN CO
- Repérer la longueur de la ligne
- POUR I DE 1 A longueur ligne
  - Lire le caractère en position I
  - SI c'est un guillemet
    - ALORS
      - SI l'indicateur est positionné à 1
        - ALORS
          - Le positionner à 0
        - SINON
          - Le positionner à 1
      - FIN SI
    - FIN SI
    - SI l'indicateur est positionné à 0
      - ALORS
        - SI le caractère est alphabétique et minuscule
          - ALORS
            - Le mettre en majuscule
        - FIN SI
      - FIN SI
      - Sauvegarder le caractère étudié
    - FIN POUR
  - FIN

## L'ORDINOGRAMME

A partir de l'algorithme précédent et en étendant l'étude sur tout un programme Basic sauvegardé en ASCII, nous pouvons élaborer l'ordinogramme suivant :







## LE PROGRAMME

De cet ordigramme complet, en tenant compte des branchements et en tenant compte de la demande de confirmation du travail à effectuer, on peut écrire le programme suivant :

```

10 REM *****
20 REM *   PROGRAMME DE MISE EN   *
30 REM *   MAJUSCULE DES VARIABLES *
40 REM *   D'UN PROGRAMME BASIC   *
50 REM *   SAUVE SOUS FORME ASCII *
60 REM *****
70 MODE 2
80 REM
90 REM *** ACQUISITION DU FICHER ***
100 INPUT "Nom du fichier qu'il faut tra
nsformer";NOMFICS
110 PRINT
120 PRINT CHR$(24);"Ce fichier doit avoi
r ete sauve en ASCII";CHR$(24)
130 PRINT
140 FOR I = 1 TO 20
150   AS=""
160 NEXT I
170 PRINT "CONFIRMEZ VOUS PRINT (O/N)"
180 WHILE AS = ""
190   AS = INKEY$
200 WEND
210 IF AS <> "O" AND AS <> "o" THEN STOP
220 REM
230 REM *** OUVERTURE DES FICHER ***
240 OPENIN NOMFICS
250 OPENOUT "MAJUSC.ASC"
260 WHILE NOT EOF
270 REM *** LECTURE LIGNE ***
280   LINE INPUT#9,LIGNES
290   PRINT LIGNES
300 REM *** INITIALISATION DES VARIABLES
***
310   INDICAT = 0
320   LIGNTEMP$ = ""
330   LONGLIGNE = LEN(LIGNES)
340   FOR I = 1 TO LONGLIGNE
350     CARAC$ = MID$(LIGNES,I,1)
360     IF CARAC$ <> CHR$(34) THEN GOTO
380:REM RECHERCHE DE GUILLEMETS
370     IF INDICAT = 1 THEN INDICAT = 0
ELSE INDICAT = 1:REM INDICATEUR BASCULA
NT A CHAQUE RENCONTRE DE GUILLEMETS

```

```
380         IF INDICAT <> 0 THEN GOTO 420:R
EM NOUS NE SOMMES PAS DANS UNE CHAINE DE
  CARACTERES DESTINES A L'AFFICHAGE
390         IF ASC(CARAC$) <= 96 OR A
SC(CARAC$) >= 123 THEN GOTO 420:REM CARA
CTERE MINUSCULE?
400         CARAC$ = CHR$(ASC(CARA
CS) - 32):REM OUI ALORS TRANSFORMER
410 REM *** RECONSTITUTION LIGNE ***
420         LIGNTEMP$ = LIGNTEMP$ + CARAC$
430         NEXT I
440 REM *** SAUVEGARDE LIGNE TERMINEE **
*
450         PRINT LIGNTEMP$
460         PRINT#9,LIGNTEMP$
470 WEND
480 REM *** FIN DU TRAITEMENT DU PROGRAM
ME ***
490 REM *** FERMETURE DES FICHIERS ***
500 CLOSEOUT
510 CLOSEIN
520 MODE 2
530 PRINT "Traitement termine"
540 PRINT
550 PRINT "Le fichier modifie se nomme M
AJUSC.ASC"
560 PRINT "Il est pret a etre traite par
le programme de recherche"
570 PRINT
580 END
```

- Lignes 100 à 210 : Acquisition et confirmation du nom du fichier à traiter.
- Lignes 240 à 280 : Lecture du fichier ligne par ligne.
- Ligne 350 : Extraction d'un caractère.
- Ligne 360 : Test de guillemets (CHR\$(34)) et affectation de l'indicateur en ligne 370.
- Ligne 390 : Test de caractère minuscule pour transformation en ligne 400.
- Ligne 420 : Reconstitution de la ligne au fur et à mesure.
- Lignes 500 à fin : Fermeture des fichiers et affichage de la fin.



## UTILISATION

Avant de lancer cet utilitaire, vous devez vous assurer que votre programme à modifier est sauvegardé en ASCII.

Puis vous exécutez le programme de conversion, en répondant aux questions.

Vous verrez défiler les lignes à l'écran vous signalant que le programme n'est pas planté, et, en fin de travail, le fichier transformé est fermé, et nommé MAJUSC.ASC. Il s'agit d'un fichier ASCII sur lequel il sera possible d'effectuer d'autres modifications, notamment le remplacement de nom de variables.

Notons que le fichier d'origine n'est pas modifié et pourra éventuellement être réutilisé en cas de doute.

## II. Le remplaceur

### L'ALGORITHME

Après avoir utilisé le majusculeur, il est possible de remplacer dans un logiciel Basic une variable, un groupe de mots, ou encore une instruction grâce au programme que nous vous proposons maintenant.

Dans ce programme, nous allons utiliser une instruction peu connue du BASIC LOCOMOTIVE : **INSTR**.

La syntaxe de cette instruction est :

**A = INSTR (CHAINE1\$,CHAINE2\$)**

Elle effectue la recherche de la chaîne de caractère CHAINE2\$ dans la chaîne CHAINE1\$.

Si aucune occurrence de CHAINE2\$ n'est trouvée, la variable A prend la valeur 0, sinon, la position du premier caractère de CHAINE2\$ dans CHAINE1\$ est transmise à A.

Connaissant la position de la chaîne de caractère à rechercher, sa longueur et la longueur de la chaîne initiale, il est possible de séparer les parties situées à gauche et à droite de la chaîne trouvée à l'aide des instructions **LEFT\$** et **RIGHT\$**.

L'algorithme de remplacement sera ainsi :

— DEBUT : REMPLACEMENT D'UNE CHAÎNE DE CARACTÈRE

CO

On suppose la chaîne de caractère trouvée à l'emplacement numéroté PLACE

FIN CO

— Calculer le nombre de caractères à gauche

CO

Un de moins que l'emplacement PLACE

FIN CO

— Calculer le nombre de caractères à droite

CO

La longueur totale, moins le nombre de caractères à gauche, moins le nombre de caractères de la chaîne à trouver

FIN CO

— Récupérer la chaîne de caractère à gauche

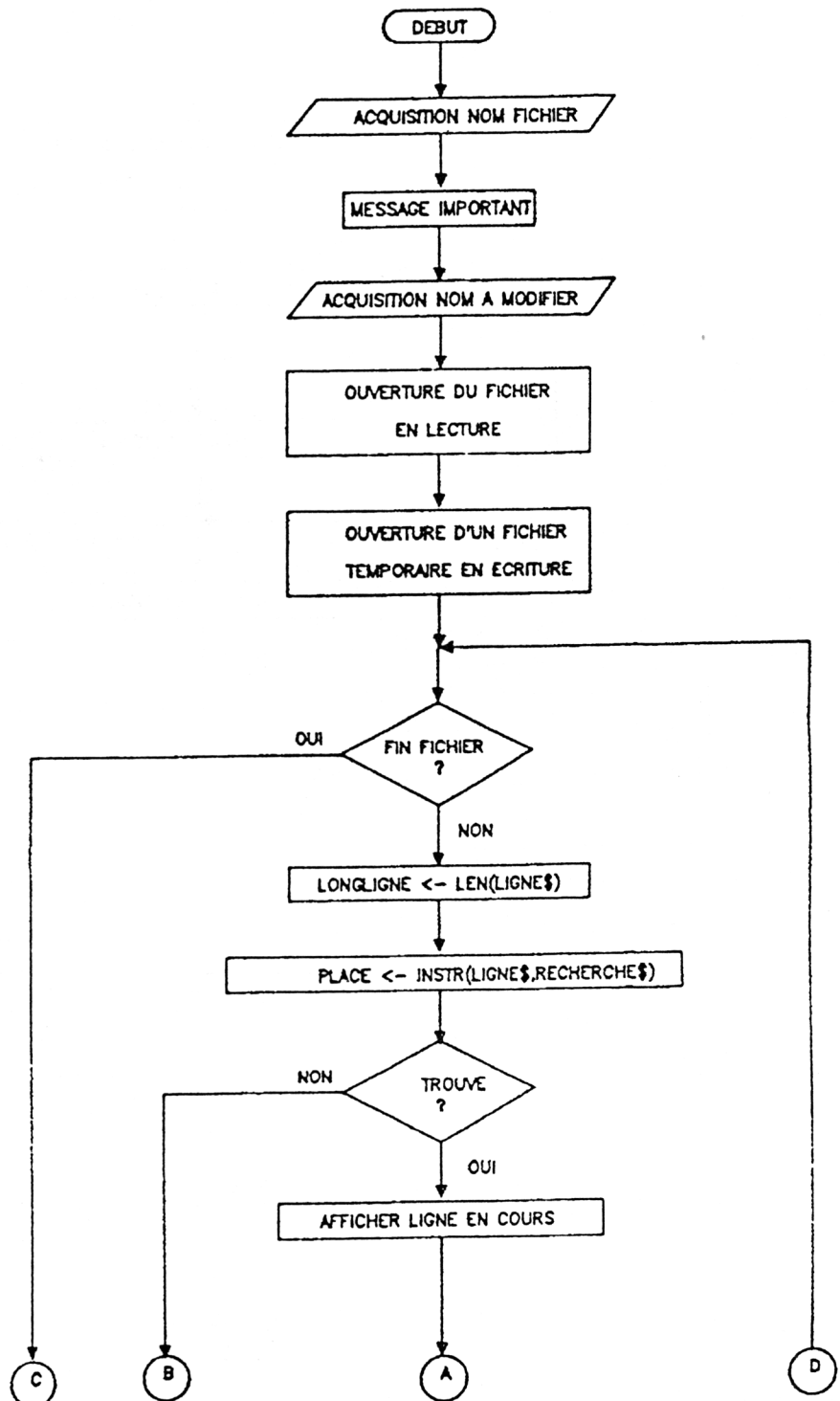
— Récupérer la chaîne de caractère à droite

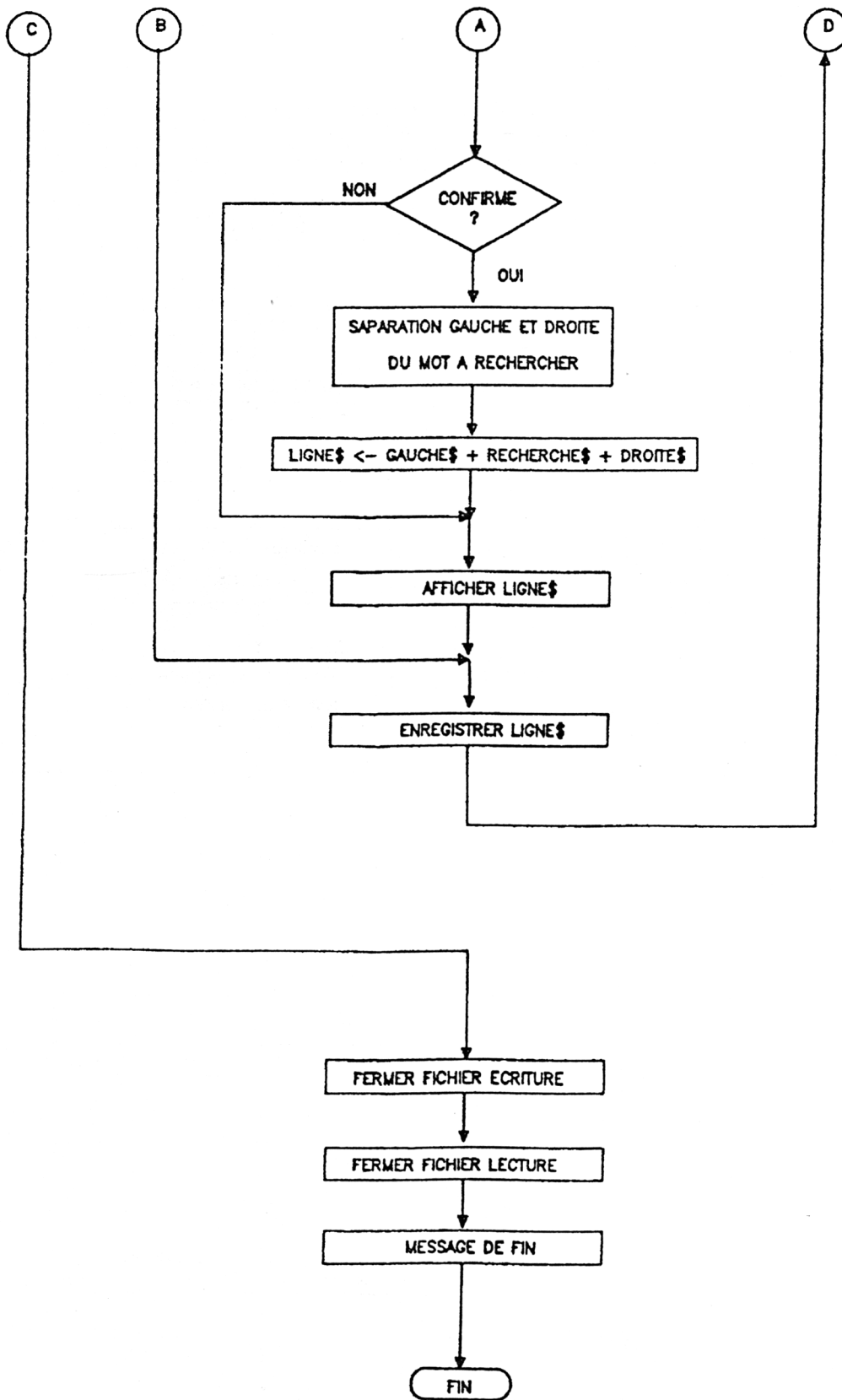
— Concatener gauche, la chaîne de remplacement et droite

— FIN

**L'ALGORIGRAMME**

En appliquant l'algorithme précédent sur toutes les lignes d'un programme Basic sauvegardé en ASCII, on obtient l'algorithme suivant :





## LE PROGRAMME

```

10 REM *****
20 REM *   PROGRAMME PERMETTANT LE   *
30 REM *     REMPLACEMENT DE MOTS   *
40 REM *     OU INSTRUCTIONS DANS   *
50 REM *     UN PROGRAMME BASIC     *
60 REM *****
70 MODE 2
80 REM
90 REM *** ACQUISITION DU FICHER ***
100 INPUT "Nom du fichier que vous desir
ez modifier";NOMFICS
110 PRINT
120 PRINT CHR$(24);"Ce fichier doit avoi
r ete sauve en ASCII";CHR$(24)
130 PRINT
140 REM
150 REM *** ACQUISITION DES MOTS ***
160 PRINT "Mot ou groupe de mots a reche
rcher dans le fichier"
170 INPUT RECHERCHES
180 LONGRECHERCHE = LEN(RECHERCHES)
190 PRINT
200 INPUT "qui doit etre remplacer par:"
;REMPACES
210 REM
220 REM *** OUVERTURE DES FICHIERS ***
230 OPENIN NOMFICS
240 OPENOUT "TEMP.ASC"
250 WHILE NOT EOF
260 REM *** LECTURE LIGNE ***
270     LINE INPUT #9,LIGNES
280     LONGLIGNE = LEN(LIGNES)
290 REM *** RECHERCHE OCCURENCE DU MOT D
ANS LA CHAINE DE CARACTERES COMPOSANT LA
LIGNE DE PROGRAMME ***
300     PLACE = INSTR(LIGNES,RECHERCHES)
310     IF PLACE = 0 THEN PRINT LIGNES:GO
TO 510:REM *** PAS TROUVE ***
320     PRINT
330 REM *** MOT TROUVE ***
340     PRINT LIGNES
350     PRINT
360     PRINT"Modification sur cette lign
e ? (O/N)"
370     INPUT MODIFS
380     IF MODIFS <> "O" AND MODIFS <> "o
" THEN GOTO 480

```

```
390 REM *** NOMBRE DE CARACTERES A GAUCH
E ***
400     GAUCHE = PLACE - 1
410 REM *** NOMBRE DE CARACTERES A DROIT
E ***
420     DROITE = LONGLIGNE - PLACE - LONG
RECHERCHE + 1
430 REM *** SEPARATION GAUCHE ***
440     GAUCHES = LEFT$(LIGNES,GAUCHE)
450 REM *** SEPARATION DROITE ***
460     DROITES = RIGHT$(LIGNES,DROITE)
470     LIGNES = GAUCHES+REPLACES+DROITE
$
480     PRINT
490     PRINT LIGNES
500     PRINT
510     PRINT#9,LIGNES
520 WEND
530 REM *** FIN DE TRAITEMENT DU PROGRAM
ME
540 REM *** FERMETURE DES FICHIERS ***
550 CLOSEOUT
560 CLOSEIN
570 MODE 2
580 PRINT"Traitement termine"
590 PRINT
600 PRINT"le fichier modifie se nomme TE
MP.ASC"
610 PRINT"apres essai, vous pourrez le r
enommer par la commande |era..."
620 PRINT
630 END
```

- Ligne 100 : Acquisition du fichier.
- Ligne 160 : Acquisition de la chaîne de caractère à remplacer.
- Ligne 200 : Entrée de la chaîne de remplacement.
- Lignes 230 et 240 : Ouverture des fichiers puis lecture d'une ligne de Basic.
- Ligne 300 : Recherche de la chaîne de caractère.
- Ligne 380 : Demande de confirmation de remplacement.
- Lignes 400 à 470 : La chaîne est éclatée puis concaténée avec la chaîne de remplacement.
- Lignes 550 à fin : Fermeture des fichiers et message de fin.

### UTILISATION

Il est nécessaire d'avoir sauvegardé le programme à modifier sous forme ASCII, ou d'utiliser le programme MAJUSC.ASC généré par le majusculeur.

On lance le programme de modification, et il suffit de répondre aux questions.

Lorsque la chaîne à rechercher est trouvée, le programme affiche la ligne à modifier et demande confirmation, pour ne pas modifier des instructions éventuelles.

Lors de l'écriture de ce programme, nous avons supposé que vous n'avez programmé qu'un nom de variable par ligne de programme. Si tel n'était pas le cas, nous vous conseillons d'ajouter la ligne suivante au programme :

```
475 GOTO 280
```

