

9/11.2.2

Premier jeu de fonctions évoluées

Dans ce paragraphe, nous vous proposons quatre fonctions destinées à améliorer le fonctionnement du traitement de texte Weka étudié à la Partie 9, chapitre 11.2. En voici la liste :

- déplacement rapide du curseur vers le début du prochain mot de la ligne courante ;
- déplacement rapide du curseur vers le début du mot précédent de la ligne courante ;
- déplacement rapide en fin de ligne courante ;
- déplacement rapide au début de la ligne courante.

Ces quatre fonctions sont exécutées par des sous-programmes écrits en Assembleur. Elles se logent immédiatement après la fin du programme de base : adresse &H8289.

Examinons en détail les diverses actions effectuées par chacune des nouvelles fonctions.

I. Déplacements rapides

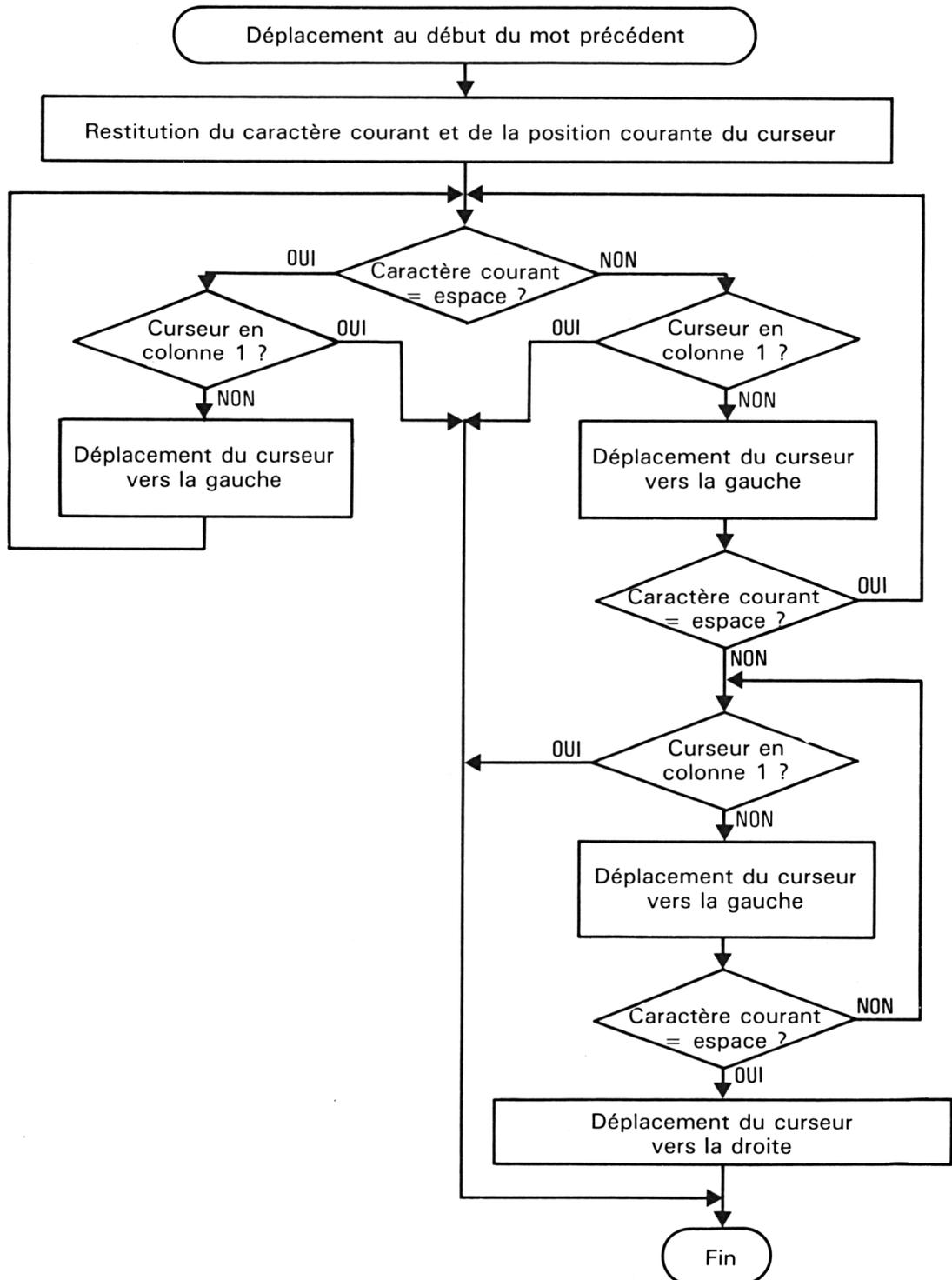
DÉPLACEMENT RAPIDE A DROITE

Lorsque l'utilisateur appuie simultanément sur les touches **Ctrl** et **A**, le curseur se déplace sur la première lettre du mot qui suit le mot courant.

Si le curseur se trouve en fin de ligne, aucun déplacement n'est effectué.

Si aucun mot ne se trouve à droite du curseur, celui-ci se déplace en fin de ligne courante.

Ces actions obéissent à la logique suivante :



Le sous-programme correspondant à cette fonction a pour nom TMDRO.

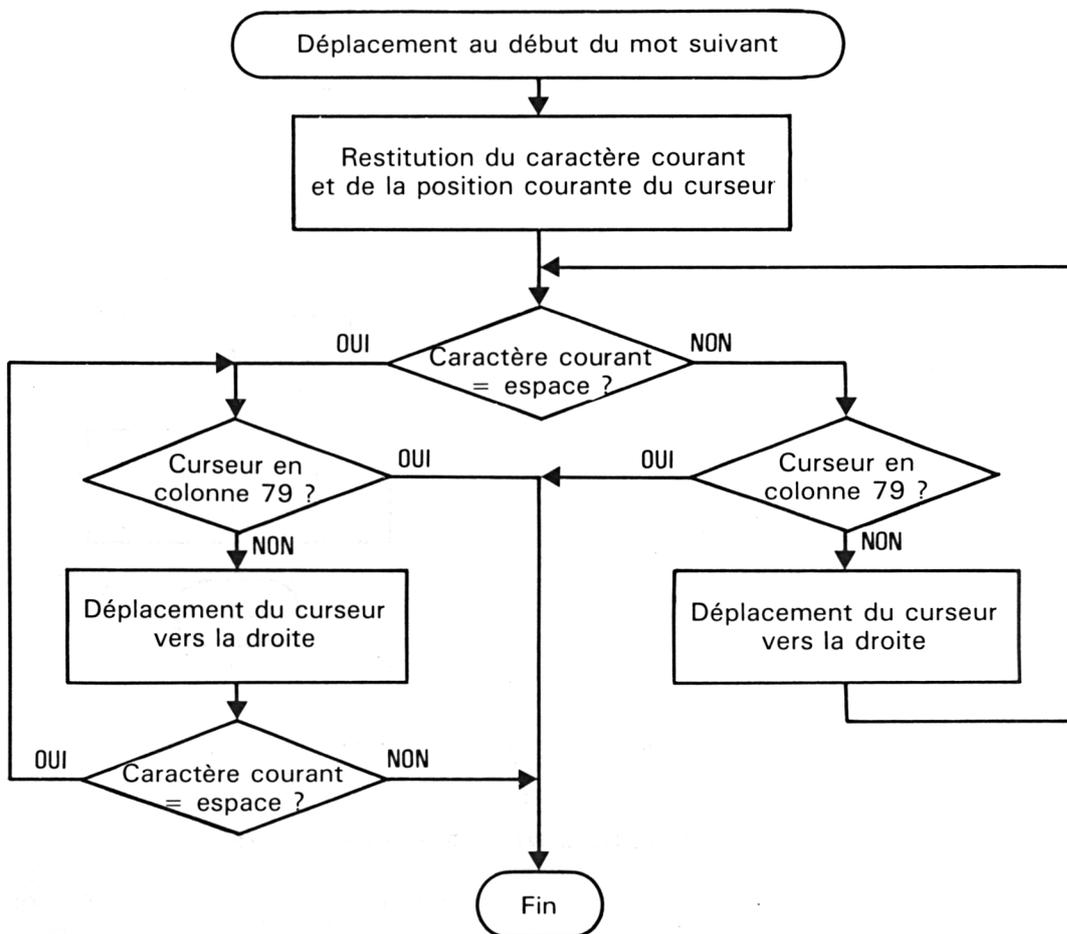
DÉPLACEMENT RAPIDE A GAUCHE

Lorsque l'utilisateur appuie simultanément sur les touches **Ctrl** et **F**, le curseur se déplace sur la première lettre du mot qui précède le mot courant.

Si le curseur se trouve en début de ligne, aucun déplacement n'est effectué.

Si aucun mot ne se trouve à gauche du curseur, celui-ci se déplace en début de ligne courante.

Ces actions obéissent à la logique suivante :

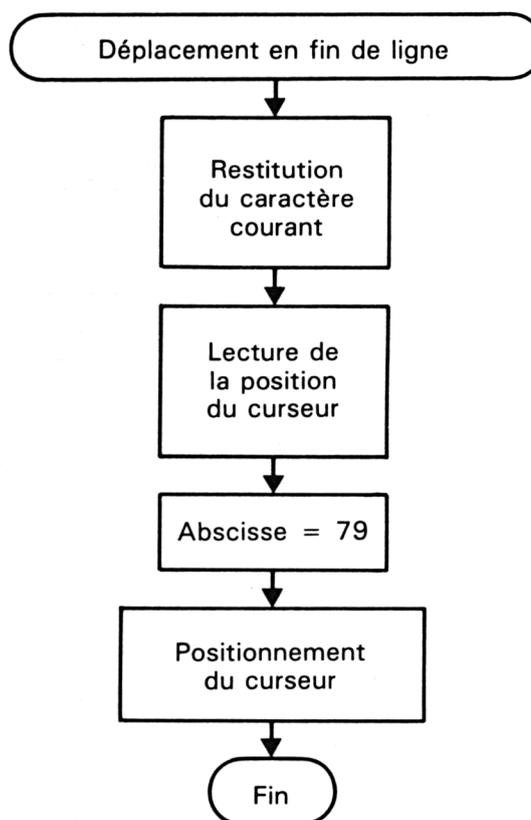


Le sous-programme correspondant à cette fonction a pour nom TMGAU.

DÉPLACEMENT EN FIN DE LIGNE

Lorsque l'utilisateur appuie simultanément sur les touches **Ctrl** et **Right**, le curseur se déplace sur le dernier caractère de la ligne courante.

Cette action met en œuvre des instructions qui obéissent à la logique suivante :

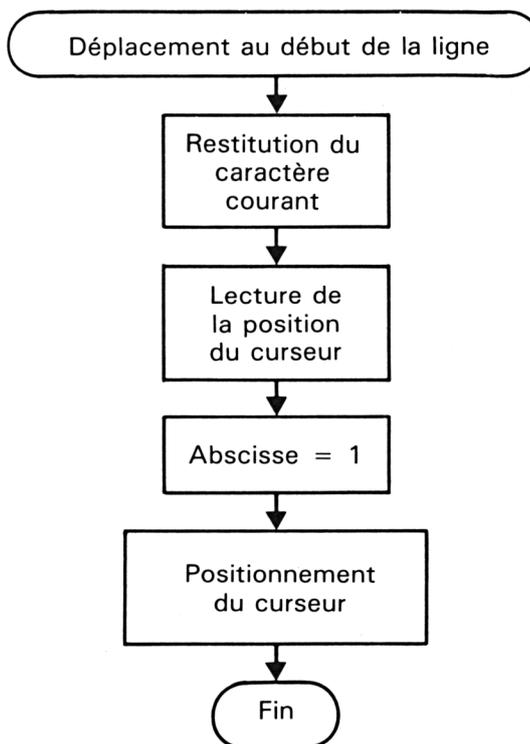


Le sous-programme correspondant à cette fonction a pour nom TFINLI.

DÉPLACEMENT AU DÉBUT DE LA LIGNE

Lorsque l'utilisateur appuie simultanément sur les touches **Ctrl** et **Left**, le curseur se déplace sur le premier caractère de la ligne courante.

Cette action met en œuvre des instructions qui obéissent à la logique suivante :



Le sous-programme correspondant à cette fonction a pour nom TDEBLI.

Le programme Assembleur qui contient ces quatre fonctions a été écrit sous ZEN. En voici le listing :

```

1          ORG 8289H
2          LOAD 0289H
3          ;-----
4          ;Suite 1 Traitement de texte WEKA
5          ;-----
6          ;Fonctions rajoutees:
7          ;Ctrl A : Deplact mot a gauche
8          ;Ctrl F : Deplact mot a droite
9          ;Ctrl Right : Fin de la ligne
    
```

```

10      ;Ctrl Left : Debut de la ligne
11      ;-----
12      ;
13 8289 FE01      CP   CTRLA
14 828B CAA3B2    JP   Z,TMGAU      ;Debut mot a gauche
15 828E FE06      CP   CTRLF
16 8290 CA0BB3    JP   Z,TMDRO      ;Debut mot a droite
17 8293 FEFB      CP   CTRLRI
18 8295 CA45B3    JP   Z,TFINLI     ;Fin de la ligne
19 8298 FEFA      CP   CTRLLE
20 829A CA54B3    JP   Z,TDEBLI     ;Debut de la ligne
21 829D C32FB2    JP   AFFICH      ;Affichage caractere
22 82A0 C31AB0    JP   ACQCAR      ;Boucle principale
23      ;-----
24      ;
25      ;-----
26      ;Deplact debut mot a gauche
27      ;-----
28      TMGAU:      EQU  $
29 82A3 CD70B2    CALL RESTIT      ;Caract courant
30 82A6 CD78BB    CALL GETCUR
31 82A9 25        DEC  H
32 82AA CD75BB    CALL SETCUR      ;Restit pas courante
33      GAUSO:      EQU  $
34 82AD CD60BB    CALL RDCHAR      ;Lect car courant
35 82B0 FE20      CP   SPACE      ;Blanc ?
36 82B2 2B1C      JR   Z,GAU2      ;Oui
37 82B4 CD78BB    CALL GETCUR
38 82B7 7C        LD   A,H
39 82B8 FE01      CP   1
40 82BA 2B3C      JR   Z,GAU4      ;Plus de deplact

```

```

41 82BC 25          DEC  H
42 82BD CD75BB     CALL SETCUR
43 82C0 CD60BB     CALL RDCHAR
44 82C3 FE20       CP   SPACE
45 82C5 2802       JR   Z,GAU0
46 82C7 1815       JR   GAU3
47                GAU0: EQU  $
48 82C9 CD60BB     CALL RDCHAR          ;Lect car courant
49 82CC FE20       CP   SPACE          ;Blanc ?
50 82CE 200E       JR   NZ,GAU3          ;Non
51                GAU2: EQU  $
52 82D0 CD78BB     CALL GETCUR
53 82D3 7C         LD   A,H
54 82D4 FE01       CP   1
55 82D6 2820       JR   Z,GAU4          ;Plus de deplact
56 82D8 25        DEC  H
57 82D9 CD75BB     CALL SETCUR
58 82DC 18EB       JR   GAU0          ;Boucle
59                GAU3: EQU  $
60 82DE CD78BB     CALL GETCUR
61 82E1 7C         LD   A,H
62 82E2 FE01       CP   1
63 82E4 2812       JR   Z,GAU4          ;Plus de deplact
64 82E6 25        DEC  H
65 82E7 CD75BB     CALL SETCUR
66 82EA CD60BB     CALL RDCHAR
67 82ED FE20       CP   SPACE          ;Blanc ?
68 82EF 20ED       JR   NZ,GAU3          ;Non
69 82F1 CD78BB     CALL GETCUR
70 82F4 24         INC  H
    
```

```

71 82F5 CD75BB          CALL SETCUR          ;Ajustement curseur
72                    GAU4:    EQU $
73 82F8 C31A80          JP    ACQCAR          ;Fin du traitement
74                    GAU1:    EQU $
75 82FB CD60BB          CALL RDCHAR
76 82FE FE01            CP    1
77 8300 28F6            JR    Z,GAU4          ;Plus de deplact
78 8302 CD78BB          CALL GETCUR
79 8305 25              DEC  H
80 8306 CD75BB          CALL SETCUR
81 8309 18BE            JR    GAU0          ;Boucle de recherche
82                    ;
83                    ;-----
84                    ;Deplact debut mot a droite
85                    ;-----
86                    TMDRO:    EQU $
87 830B CD7082          CALL RESTIT          ;Restit car courant
88 830E CD78BB          CALL GETCUR          ;Position curseur
89 8311 25              DEC  H
90 8312 CD75BB          CALL SETCUR          ;Restit pos curseur
91                    DROI0:    EQU $
92 8315 CD60BB          CALL RDCHAR          ;Lecture car courant
93 8318 FE20            CP    SPACE          ;Blanc?
94 831A 2015            JR    NZ,DROI1        ;Non
95                    DROI2:    EQU $
96 831C CD78BB          CALL GETCUR
97 831F 7C              LD    A,H
98 8320 FE4F            CP    79
99 8322 281E            JR    Z,DROI3        ;Fin de deplact
100 8324 24             INC  H
101 8325 CD75BB          CALL SETCUR

```

```

102 832B CD60BB          CALL RDCHAR
103 832B FE20           CP   SPACE
104 832D 28ED           JR   Z,DRO12           ;Boucle
105 832F 1811           JR   DRO13           ;Fin de deplact
106                      ;
107                    DRO11: EQU $
108 8331 CD78BB          CALL GETCUR
109 8334 7C             LD   A,H
110 8335 FE4F           CP   79             ;Fin de ligne ?
111 8337 2809           JR   Z,DRO13           ;Oui
112 8339 CD78BB          CALL GETCUR
113 833C 24            INC  H
114 833D CD75BB          CALL SETCUR
115 8340 18D3           JR   DRO10           ;Boucle
116                    DRO13: EQU $
117 8342 C31A80          JP   ACQCAR           ;Fin du traitement
118                      ;
119                      ;-----
120                      ;Deplact fin de la ligne
121                      ;-----
122                    TFINLI: EQU $
123 8345 CD7082          CALL RESTIT           ;Caract courant
124 8348 CD78BB          CALL GETCUR
125 834B 3E4F           LD   A,79
126 834D 67            LD   H,A
127 834E CD75BB          CALL SETCUR           ;Curs en fin de ligne
128 8351 C31A80          JP   ACQCAR           ;Fin de traitement
129                      ;
130                      ;-----
131                      ;Deplact debut de la ligne
132                      ;-----

```

```
133          TDEBLI:      EQU  $
134 8354 CD70B2          CALL RESTIT          ;Caract courant
135 8357 CD78BB          CALL GETCUR
136 835A 3E01           LD   A,1
137 835C 67            LD   H,A
138 835D CD75BB          CALL SETCUR          ;Position curseur
139 8360 C31A80          JP   ACQCAR          ;Fin traitement
140          ;
141          ;=====
142          ;   ZONE   DES   EQU
143          ;=====
144          ;
145          GETCUR:      EQU  0BB78H          ;TXT GET CURSOR
146          SETCUR:      EQU  0BB75H          ;TXT SET CURSOR
147          RDCHAR:      EQU  0BB60H          ;TXT RD CHAR
148          RESTIT:      EQU  8270H          ;Rest car cour
149          CTRLA:      EQU  01H
150          CTRLF:      EQU  06H
151          CTRLRI:      EQU  251
152          CTRLLE:      EQU  250
153          SPACE:      EQU  32
154          ;
155          AFFICH:      EQU  822FH          ;Ref 1er prog
156          ACQCAR:      EQU  801AH          ;Ref 1er prog
157          ;
158          ;
159          END
```

Comme toujours, voici la version « Chargeur Basic » correspondant au programme Assembleur listé ci-dessus :

```

1000 ' =====
1010 ' Suite 1 de l'editeur de texte WEKA
1020 ' =====
1030 '
1040 ' -----
1050 ' Memorisation des S/P Asm
1060 ' -----
1070 FOR I=&B2B9 TO &B360
1080   READ A$
1090   A=VAL("&"+A$)
1100   POKE I,A
1110 NEXT I
1120 '
1130 ' -----
1140 POKE &B051,&B9 : POKE &B052,&B2 'Modif du 1er programme
1150 ' -----
1160 CALL &B000 'Execution
1170 END
1180 '
1190 ' -----
1200 ' Donnees des S/P Assembleur
1210 ' -----
1220 DATA FE,1,CA,A3,82,FE,6,CA,B,83,FE,FB,CA,45,83,FE
1230 DATA FA,CA,54,83,C3,2F,82,33,1A,80,CD,70,82,CD,78,BB
1240 DATA 25,CD,75,BB,CD,60,BB,FE,20,28,1C,CD,78,BB,7C,FE
1250 DATA 1,28,3C,25,CD,75,BB,CD,60,BB,FE,20,28,2,18,15
1260 DATA CD,60,BB,FE,20,20,E,CD,78,BB,7C,FE,1,28,20,25
1270 DATA CD,75,BB,18,EB,CD,78,BB,7C,FE,1,28,12,25,CD,75
1280 DATA BB,CD,60,BB,FE,20,20,ED,CD,78,BB,24,CD,75,BB,C3
1290 DATA 1A,80,CD,60,BB,FE,1,28,F6,CD,78,BB,25,CD,75,BB
1300 DATA 18,BE,CD,70,82,CD,78,BB,25,CD,75,BB,CD,60,BB,FE
1310 DATA 20,20,15,CD,78,BB,7C,FE,4F,28,1E,24,CD,75,BB,CD
1320 DATA 60,BB,FE,20,28,ED,18,11,CD,78,BB,7C,FE,4F,28,9
1330 DATA CD,78,BB,24,CD,75,BB,18,D3,C3,1A,80,CD,70,82,CD
1340 DATA 78,BB,3E,4F,67,CD,75,BB,C3,1A,80,CD,70,82,CD,78
1350 DATA BB,3E,1,67,CD,75,BB,C3,1A,80,0,0,0,0,0,0

```

Assurez-vous que les données entrées sont correctes grâce au programme de checksum qui doit donner le résultant suivant :

DC 34 EE E9 23 24 BB C9 A6 59 78 FD 8D BF

Si une des données affichées par le programme de checksum n'est pas identique à celle de même rang dans la liste ci-dessus, la ligne de DATA correspondante contient une ou plusieurs erreurs de frappe...

Pour implanter les quatre fonctions dont nous venons de parler, il faut procéder de la manière suivante :

- si vous utilisez un Assembleur :
 - entrez les codes opératoires donnés dans le premier listing,
 - implantez le programme de base en mémoire,
 - placez les valeurs &H89 et &H82 dans les octets d'adresse &H8051 et &H8052, ceci pour effectuer un débranchement dans la boucle principale,
 - activez le programme en tapant, sous Basic Call &8000.
- si vous préférez utiliser le chargeur Basic :
 - chargez en mémoire le premier chargeur (Partie 9, Chap. 11.2.1 page 30),
 - supprimez la ligne 1080,
 - sauvegardez puis exécutez le chargeur ainsi modifié,
 - chargez en mémoire le second chargeur (voir Partie 9, Chap. 11.2.2 page 11),
 - exécutez ce second chargeur. Vous vous trouvez dans le traitement de texte.

II - Résumé des fonctions disponibles

Les commandes désormais disponibles dans le traitement de texte Weka sont les suivantes :

- touches flèches pour déplacer le curseur dans toutes les directions,
- **Carriage Return** pour passer à la ligne,
- **Ctrl Z** pour effectuer un scrolling d'une ligne vers le haut,
- **Ctrl W** pour effectuer un scrolling d'une ligne vers le bas,
- **CLR** pour effacer le caractère courant,
- **DEL** pour effacer le caractère à gauche du caractère courant,
- **Ctrl A** pour déplacer le curseur sur la première lettre du mot qui se trouve à droite du mot courant,
- **Ctrl F** pour déplacer le curseur sur la première lettre du mot qui se trouve à gauche du mot courant,
- **Ctrl Right** pour déplacer le curseur sur le dernier caractère de la ligne courante,
- **Ctrl Left** pour déplacer le curseur sur le premier caractère de la ligne courante,
- **Ctrl Q** pour quitter le traitement de texte et retourner sous Basic.