AMSTRAD

PROGRAMMEZ VOTRE TRAITEMENT DE TEXTE

JEAN-CLAUDE DESPOINE

SYBEX

AMSTRAD

PROGRAMMEZ VOTRE TRAITEMENT DE TEXTE

DANS LA MÊME COLLECTION

Amstrad jeux d'action, P. Monsaut
Amstrad premiers programmes, R. Zaks
Amstrad 56 programmes, S. R. Trost
Amstrad guide du BASIC et de l'AMSDOS, J.-L. Gréco/M. Laurent
Amstrad exploré, J. Braga
Amstrad programmation en assembleur, G. Fagot-Barraly
Amstrad guide du graphisme, J. Wynford
Amstrad CP/M 2.2, A. d'Hardancourt
Amstrad astrologie, numérologie, biorythmes, P. Bourgault
Amstrad graphisme en trois dimensions, T. Lachand-Robert
Amstrad Multiplan, Amstrad
Amstrad CP/M plus, A. d'Hardancourt
Amstrad Astrocalc, G. Blanc/P. Destrebecq
Amstrad gagnez aux courses, J.-C. Despoine
Amstrad créer de nouvelles instructions, J.-C. Despoine
Amstrad Locoscript, B. Le Dû
Amstrad mise au point des programmes BASIC, C. Vivier/Y. Jacob
Amstrad jeux en assembleur, E. Ravis
Amstrad techniques de programmation des jeux, G. Fagot-Barraly
Amstrad routines en assembleur, J.-C. Despoine
Amstrad mieux programmer en assembleur, T. Lachand-Robert
Amstrad jeux de réflexion, G. Fagot-Barraly
Amstrad programmes en langage machine, S. Webb
Amstrad gérez votre portefeuille boursier, J.-C. Despoine/F. Pierre
Amstrad PCW 8256/8512 guide du BASIC et de Jetsam, J.-L. Gréco/M. Laurent
Amstrad guide de Logo, A. d'Hardancourt (à paraître)
Amstrad introduction à la programmation en assembleur sous CP/M, A. d'Hardancourt (à paraître)
Amstrad Startext (logiciel de traitement de texte (à paraître))

JEAN-CLAUDE DESPOINE

AMSTRAD

PROGRAMMEZ VOTRE TRAITEMENT DE TEXTE

Paris • Berkeley • Düsseldorf

ISBN 2-7361-0221-5.

S O M M A I R E

MODE D'EMPLOI DU TRAITEMENT DE TEXTE

■ LES COMMANDES CURSEUR

Curseur vers la droite

Déplacement du curseur d'une position vers la droite, sans modification du texte (les dépassements de fin de texte par simple déplacement du curseur sont acceptés et entérinés).

Curseur vers la gauche

Déplacement du curseur d'une position vers la gauche, sans modification du texte.

Curseur vers le haut

Déplacement du curseur d'une ligne vers le haut.

Curseur vers le bas

Déplacement du curseur d'une ligne vers le bas (l'écran de travail est constitué de 255 lignes dont 22 peuvent être affichées simultanément).

SHIFT + curseur vers la droite

Renvoi du curseur à la colonne 80 de la ligne courante, ou après le dernier caractère de cette ligne s'il s'agit de la dernière du texte. Si le curseur s'y trouve déjà, un coup de cloche est envoyé et la commande reste sans effet.

SHIFT + curseur vers la gauche

Renvoi du curseur à la colonne 1 de la ligne courante. Si le curseur s'y trouve déjà, un coup de cloche est envoyé et la commande reste sans effet.

SHIFT + curseur vers le haut

Affichage de la première page de l'écran de travail, et positionnement du curseur au début de cette page.

SHIFT + curseur vers le bas

Affichage de la dernière page de l'écran de travail, et posi-

tionnement du curseur à la fin du texte. Notons que cette commande ne provoque pas forcément de remplissage de l'écran : une page représentant 1 760 caractères (22 lignes de 80 colonnes), si par exemple 1 900 caractères ont été écrits, l'appel de cette commande ne provoquera l'affichage que des 140 derniers et le reste de l'écran restera vierge.

■ LES CARACTÈRES

Les caractères accentués peuvent être obtenus grâce au pavé numérique, de la manière suivante :

Touche	"." :	î
Touche	"0" :	ô
Touche	"1" :	â
Touche	"2" :	ï
Touche	"3" :	û
Touche	"4" :	à
Touche	"5" :	ç
Touche	"6" :	ù
Touche	"7" :	é
Touche	"8" :	è
Touche	"9" :	ê

Les crochets " [" et "] ", les accolades " { " et " } " et l'arobas " @ " sont accessibles, y compris à l'impression.

SHIFT + barre d'espacement

Permet d'écrire un blanc forcé, symbolisé par un petit signe en forme de vague. A l'affichage ou à l'impression, ces blancs forcés sont traités comme des caractères à part entière, et ne doivent en aucun cas être considérés comme des séparateurs de mots. Ils peuvent entre autres être utiles pour réserver des espaces blancs dans un texte.

Sont considérés comme des séparateurs de mots les espaces inutiles (symbolisés par un point), les blancs, les codes de contrôle, ainsi bien sûr que les codes de fin de paragraphe.

■ LE MODE INSERTION

Le mode insertion est obtenu grâce à la touche COPY, et sa suppression grâce à une seconde frappe de cette même touche. La mise en service du mode insertion est signalée par l'affichage de l'abréviation INS dans le petit carré de la bande inférieure de l'écran.

Si, au moment de l'insertion, le curseur pointe un espace inutile, le caractère est affiché normalement ; dans tous les autres cas, le reste du texte est décalé, de manière que le caractère puisse être affiché dans l'espace ainsi libéré. Le code de fin de paragraphe est obligatoirement inséré (même si le mode insertion n'est pas en service), s'il est appelé alors que le curseur ne se trouve pas en fin de texte.

Attention, toute insertion dans le texte alors que la RAM de rangement est pleine aura pour effet de " chasser " le dernier caractère.

■ LES TOUCHES " CLR", " DEL " et " SHIFT + CLR "

CLR

Efface le caractère pointé par le curseur, y compris s'il s'agit d'un espace inutile, et décale le reste du texte pour combler le vide.

DEL

Efface le caractère situé avant le curseur, sans décaler le reste du texte. Si le curseur pointe à ce moment-là sur la fin du texte, le pointeur de fin de texte est décrémenté de 1.

SHIFT + CLR

Cette commande a pour effet de supprimer toute la ligne sur laquelle se trouve le curseur, le reste du texte étant alors décalé de 80 pour combler l'espace vacant. Cette commande n'est pas acceptée si le curseur pointe sur la fin du texte.

■ LES TOUCHES "ENTER" et "SHIFT + ENTER"

SHIFT + ENTER

Cette commande inscrit le code de fin de paragraphe ou de phrase (symbolisé par un triangle noir), et envoie le curseur au début de la ligne suivante. On peut naturellement utiliser ce code seul au début d'une ligne pour indiquer qu'elle doit simplement être sautée. Il faut également noter qu'à l'intérieur du texte, ce code de fin de paragraphe est obligatoirement inséré, même si le mode insertion n'est pas en service.

ENTER

Cette commande envoie le curseur au début de la ligne suivante. Elle n'a aucune autre fonction et ne doit surtout pas être confondue avec la précédente (si par exemple vous tapez un mot, puis ENTER, puis un autre mot, le programme considérera qu'à l'affichage ou à l'impression, ces deux mots doivent être écrits l'un à la suite de l'autre).

■ LES CODES DE CONTRÔLE

Les codes de contrôle permettent de modifier à n'importe quel moment le format de sortie du texte, ou de solliciter les différentes options de l'imprimante.

Un code de contrôle doit toujours être composé de 5 caractères dont le premier, obtenu grâce à CTRL + Z (qui provoque l'affichage d'un C en vidéo inverse), n'est qu'un simple signal. Voici les 15 codes de contrôle possibles.

CTRL + Z + 01xx

Saut de xx lignes. Ce code est évidemment intéressant dans la mesure où il évite d'avoir à utiliser des codes de fin de paragraphe lorsque l'on souhaite sauter plusieurs lignes, et permet ainsi de gagner de la place en mémoire.

CTRL + Z + 02xx

Fixe la marge gauche à la colonne xx.

CTRL + Z + 03xx

Fixe la marge droite à la colonne XX.

Chaque fois qu'une marge est fixée par un code de contrôle le programme vérifie que le nombre de caractères par ligne qui en découle reste supérieur ou égal à 15. Si ce n'est pas le cas, le code est ignoré.

Dans certains cas, cela implique que les poses de marge doivent être faites dans un ordre particulier : imaginons par exemple que vous ayez, à un moment du texte, une marge gauche à 40, une marge droite à 60, et que vous vouliez passer à une marge gauche de 5 et une marge droite de 30. Il est bien évident que si vous commencez par fixer la marge droite, le programme la refusera puisqu'il va comparer cette nouvelle marge avec *l'ancienne gauche*. Dans ce cas précis, il faudra donc d'abord fixer la nouvelle marge gauche, et seulement ensuite la nouvelle marge droite. Il faut également signaler que le nombre maximal de caractères par ligne est 80.

Tous les codes suivants sont réservés à l'imprimante, et ignorés lors de la sortie écran.

CTRL + Z + 1000	Sélectionne l'option NLQ.
CTRL + Z + 1100	Annule l'option NLQ.
CTRL + Z + 1200	Sélectionne l'option italique.
CTRL + Z + 1300	Annule l'option italique.
CTRL + Z + 1400	Sélectionne l'option caractères gras.
CTRL + Z + 1500	Annule l'option caractères gras.
CTRL + Z + 1600	Sélectionne l'option soulignement
CTRL + Z + 1700	Annule l'option soulignement.
CTRL + Z + 1800	Sélectionne l'option caractères condensés.
CTRL + Z + 1900	Annule l'option caractères condensés.
CTRL + Z + 1:00	Sélectionne l'option double largeur.
CTRL + Z + 1;00	Annule l'option double largeur.

Les caractères double largeur ou condensés ne sont pas pris en considération lors des calculs relatifs à la justification du texte. Ces caractères particuliers ne doivent donc être utilisés qu'à bon escient. Cela ne devrait plus vous poser de problème lorsque vous aurez un peu de pratique. Dans un premier temps, n'hésitez pas à faire des expériences en essayant différents cas de figure.

Les poses de marge rencontrées lors de la sortie d'une ligne ne sont entérinées qu'*après* que cette ligne a été sortie. Si l'on souhaite par exemple que les marges soient modifiées à partir d'une ligne L, il est nécessaire que les codes de contrôle correspondants soient situés *à l'intérieur de la ligne L-1*. Là encore, un peu de pratique sera sans doute nécessaire.

Le processus est le même pour les sauts de ligne, qui ne sont exécutés qu'après la sortie de la ligne où ils ont été rencontrés, tandis que les codes de contrôle imprimante le sont au fur et à mesure de l'impression.

Tout cela est bien sûr parfaitement logique, mais il faut savoir qu'une utilisation optimale du traitement de texte ne sera possible qu'avec une bonne maîtrise des codes de contrôle, dont l'incidence dépend parfois de l'endroit où ils sont placés dans le texte.

■ LE MODE MENU

Le mode MENU s'obtient grâce à la petite touche ENTER du pavé numérique. Les différentes options s'affichent alors dans la bande inférieure de l'écran. (Il est toujours possible, à n'importe quel moment, de revenir à l'écran de travail en appuyant une deuxième fois sur la petite touche ENTER ou en tapant un chiffre illégal en réponse à une interrogation.)

Les quatre choix possibles de ce menu principal sont les suivants (chacun d'entre eux branchant sur des sous-menus et tout choix illégal provoquant un retour immédiat à l'écran de travail).

LES SUPPRESSIONS

- Une suppression totale du texte provoque une réinitialisation de l'écran et de la mémoire (cela a le même effet que si le programme venait d'être lancé). Il est également possible de n'effacer qu'une partie du texte, cette partie étant alors définie par rapport à la position du curseur au moment où a eu lieu l'appel du mode MENU.
- Une suppression " À PARTIR DU CURSEUR " efface le caractère pointé par ce dernier ainsi que tous ceux qui suivent.
- Une suppression " JUSQU'AU CURSEUR " efface toutes les lignes de l'écran de travail situées avant celle où se trouve le curseur, l'affichage de l'écran étant naturellement modifié en

conséquence. Attention : avant une suppression " JUSQU'AU CURSEUR ", celui-ci doit être positionné en colonne 1. Si tel n'est pas le cas, un message d'erreur s'affiche et il faut alors taper la petite touche ENTER pour retourner à l'écran de travail.

LA DÉFINITION DU FORMAT, CONSTITUÉ DE CINQ PARAMÈTRES

- La marge gauche et la marge droite (ces poses de marges n'empêchant évidemment pas que ces dernières soient modifiées en cours de texte grâce aux codes de contrôle).
- La marge haute, qui représente le nombre de lignes à sauter à chaque début de page lors des sorties.
- L'interligne entre chaque ligne d'écriture. Si ce paramètre est à 1, les sorties se font normalement ; s'il est à 2, une ligne est sautée entre chaque ligne d'écriture.
- Le nombre de lignes par page (maximum 62). Lors de la sortie écran, la fin d'une page est signalée par un trait en pointillé. Lors de la sortie imprimante, la feuille est poussée hors du chariot quand le nombre de lignes spécifié est atteint, et une nouvelle feuille est attendue.

Chaque fois qu'un format est défini, le programme propose sa sauvegarde. Il faut alors lui donner un nom et les cinq paramètres sont stockés sous forme binaire.

Un format sauvegardé peut être chargé à n'importe quel moment grâce à l'option CHARGEMENT.

Lorsqu'un texte est sauvegardé, les cinq paramètres du format sont sauvegardés en même temps, et dans le même fichier. L'inverse est naturellement vrai lors du chargement d'un texte.

Le format par défaut est le suivant :

Marge gauche = 2
Marge droite = 78
Marge haute = 1
Interligne = 1
Nombre de lignes par page = 62

LES SORTIES ET LES ENTRÉES

La sortie écran

Si cette option est choisie, il est nécessaire de préciser à partir de quelle page doit se faire l'affichage (attention, il ne s'agit pas cette fois des pages de l'écran de travail, mais bien des pages sous leur forme définitive, donc justifiées). La sortie écran étant en principe beaucoup sollicitée (généralement pour vérifier la présentation), la possibilité d'affichage à partir d'une page particulière évite d'avoir à faire défiler tout le texte, alors que seule, par exemple, la dixième page doit être vérifiée.

Lorsque le numéro de page a été défini, l'écran devient blanc, et il faut appuyer sur la barre d'espacement pour que le texte commence à défiler (tout relâchement de la barre fige l'écran en l'état).

Une fois la fin du texte atteinte, un message demande d'appuyer sur la petite touche ENTER, ce qui provoquera un retour à l'écran de travail (ce dernier étant restitué dans le même état qu'au moment de l'appel d'affichage).

Si le numéro de page spécifié dépasse le nombre de pages du texte, le message évoqué ci-dessus s'affichera immédiatement.

Si le numéro de page spécifié est important (affichage à partir de la dixième page, par exemple), il peut s'écouler un certain temps entre le moment où l'on presse la barre d'espacement et celui où commence l'affichage (le programme étant de toute façon obligé d'effectuer les calculs de justification à partir de la première page).

Il est possible d'arrêter l'affichage pour retourner à l'écran de travail à n'importe quel moment. Il suffit pour cela d'appuyer sur la petite touche ENTER.

La sortie imprimante

Le principe est le même que pour la sortie écran, seule l'interruption d'impression (petite touche ENTER) diffère quelque peu. Il peut être nécessaire, en effet, de maintenir appuyée cette touche quelques instants avant que n'apparaisse le message d'interruption. D'autre part, l'imprimante continuera de travailler, même après le retour à l'écran de travail, jusqu'à ce que son buffer soit vide (sauf naturellement si vous l'éteignez).

Si un mot trop long pour entrer dans les marges spécifiées est rencontré, l'affichage ou l'impression s'arrête immédiatement et un message d'erreur s'affiche, toute justification étant alors impossible.

Les imprimantes sont parfois des machines capricieuses. La prudence exige donc une sauvegarde systématique du texte avant toute impression.

Si l'imprimante n'est pas ON LINE, alors que l'impression est demandée, le programme bouclera jusqu'à ce qu'il y soit remédié.

Sauvegarde d'un texte

Cette option permet la sauvegarde du texte en mémoire (ainsi que du format en cours), sous la forme d'un dossier ASCII. Un nom doit être fourni, composé de 8 caractères au maximum. Lorsque la sauvegarde est terminée, la main est rendue à l'écran de travail.

Chargement d'un texte

Le texte (appelé par son nom) est chargé en RAM, après quoi sa première page est affichée tandis que le curseur est positionné en ligne 1, colonne 1 (il est naturellement possible de sauter immédiatement en fin de texte grâce à SHIFT + CURSEUR BAS).

Une demande de chargement ou de sauvegarde alors qu'il n'y a pas de disquette dans le lecteur provoque un message d'erreur et un arrêt du programme. Cela peut être particulièrement gênant si le texte sur lequel on travaille n'a pas été sauvegardé. Il faut alors, dans ce cas, utiliser la *procédure de secours* présentée plus loin.

LES UTILITAIRES

Le retour au BASIC

Cette commande provoque un "CALL 0", et réinitialise la machine. Pas question, naturellement, de récupérer quoi que ce soit ensuite. A utiliser, donc, avec prudence...

Voir le contenu d'une disquette

Cette commande affiche le contenu de la disquette, le contenu initial de l'écran de travail étant restitué lorsque l'on presse une touche.

■ DIVERS

- Lors de la saisie du texte sur écran de travail, le programme ne se préoccupe en rien des coupures de mot qui peuvent se produire en fin de ligne. Il va de soi que ces coupures disparaissent à la justification.

  Vous pouvez donc frapper votre texte sans discontinuer et sans vous faire de souci, même si des mots se retrouvent écrits "à cheval" sur deux lignes.

- La capacité maximale de la RAM de stockage du texte est de 20 400 caractères, y compris les espaces inutiles. Cela correspond bien sûr aux 255 lignes de 80 colonnes chacune de l'écran de travail (tout dépassement de la ligne 255 en écran de travail est par conséquent rigoureusement impossible).

  A titre indicatif, cette capacité mémoire représente en moyenne 5 à 10 pages justifiées, de 62 lignes chacune (ce chiffre étant susceptible de varier fortement en fonction du format, des sauts de ligne, etc.).

- En cas d'imprévu, comme par exemple une interruption accidentelle du programme (réponse erronée lors d'une saisie du menu, pas de disquette dans le lecteur lors d'un chargement ou d'une sauvegarde, etc.), le texte éventuellement en mémoire peut être récupéré en faisant un RUN 1990 (voir listing BASIC). L'utilisateur doit alors indiquer le nom du texte, qui est sauvegardé normalement ; il faut ensuite réinitialiser la machine, relancer le programme et recharger le texte si l'on souhaite continuer à travailler dessus (la réinitialisation est nécessaire pour éviter tout problème ultérieur).

Si vous vous rendez compte, en essayant de taper l'instruction évoquée précédemment, que rien ne s'affiche sur l'écran ou que les lettres disparaissent immédiatement, c'est que vous jouez vraiment de malheur, et que l'interruption accidentelle a eu lieu au mauvais moment. Rassurez-vous, rien n'est perdu pour autant puisqu'il vous suffira de taper (à l'aveuglette si nécessaire) " WINDOW #0,1,8,1,25 " + retour chariot, pour que les choses se rétablissent.

■ PROCÉDURE DE PREMIÈRE UTILISATION

Avant de pouvoir utiliser le programme, il est nécessaire de créer le fichier contenant la partie assembleur. Pour ce faire, il faut donc dans un premier temps écrire le programme de chargement correspondant et le lancer ; le fichier ASSEMBLE sera alors rangé sur cassette ou disquette (pour les 664, 6128 et 464 sans lecteur de disquette, deux fichiers seront créés – se reporter au chapitre traitant de ce sujet). Il faut ensuite écrire le programme BASIC de gestion du traitement de texte (attention aux erreurs), et le sauvegarder en lui donnant un nom, cela va de soi. Il est fortement conseillé d'effectuer cette sauvegarde sur la même disquette que la précédente.

L'utilisation du programme est dès lors extrêmement simple : il suffit d'allumer la machine et de faire un RUN " nom du programme ".

LE PROGRAMME ASSEMBLEUR

Nous avons essayé d'annoter de la manière la plus précise possible le listing assembleur du programme, et une lecture attentive devrait vous permettre d'en comprendre la logique sans trop de mal.

Il est bien certain, néanmoins (et ceux qui s'y sont déjà essayé s'en sont certainement rendu compte), qu'expliquer un programme important dans ses moindres détails représente un exercice extrêmement difficile, pour ne pas dire impossible, tant il est vrai que la démarche informatique n'est qu'un perpétuel aller et retour entre une vision synthétique du problème à résoudre et une décomposition analytique en une multitude de sous-problèmes.

Les lecteurs désireux de comprendre ce programme, sans se contenter de l'utiliser, devront donc faire leur part de travail, mais nous sommes tout à fait persuadés qu'ils en tireront un immense profit, sous réserve, cela va de soi, qu'ils soient bien familiarisés avec le CPC et possèdent les connaissances de base de l'assembleur. Précisons enfin qu'à notre avis, il est éminemment souhaitable d'utiliser un certain temps le traitement de texte en tant que tel, pour bien s'y habituer, avant de s'attaquer à l'étude du programme proprement dit.

Les informations présentées dans ce chapitre sont complémentaires de celles du listing assembleur et devraient considérablement vous faciliter la tâche (le rôle de chacun des sous-programmes, en particulier, y est décrit d'une manière formelle, ainsi que l'état des registres en entrée et en sortie). N'hésitez pas, donc, à vous y référer sans cesse.

```
      ** INITIALISATION **

1     8E94  21983A   LD HL,3A98    ;* &FF EN MEMOIRE DE 15000 A 36120
2     8E97  018052   LD BC,5280    ;
3     8E9A  36FF     LD (HL),FF   ;
4     8E9C  23       INC HL       ;
5     8E9D  0B       DEC BC       ;
6     8E9E  78       LD A,B       ;
7     8E9F  B1       OR C         ;
8     8EA0  20F8     JRNZ 8E9A    ;
9     8EA2  21983A   LD HL,3A98   ;* AFFICHAGE ECRAN
10    8EA5  0150C0   LD BC,C050   ;adresse ecran debut ligne 2
11    8EA8  11E006   LD DE,06E0   ;nombre caracteres ligne 2 a 23
12    8EAB  CDC194   CALL 94C1    ;afficher DE caracteres
13    8EAE  CDCA95   CALL 95CA    ;afficher curseur
14    8EB1  3E2F     LD A,2F      ;* REDEFINITION DES TOUCHES
15    8EB3  0600     LD B,00      ;
16    8EB5  CD2DBB   CALL BB2D    ;0 pour SHIFT + espace
17    8EB8  3E12     LD A,12      ;
18    8EBA  0601     LD B,01      ;
19    8EBC  CD2DBB   CALL BB2D    ;1 pour SHIFT + ENTER
20    8EBF  3E10     LD A,10      ;
21    8EC1  0602     LD B,02      ;
22    8EC3  CD2DBB   CALL BB2D    ;2 pour SHIFT + CLR
23    8EC6  C9       RET          ;
```

```
      ** SAISIE CLAVIER **

24    8ECC  CD1BBB    CALL BB1B     ;
25    8ECF  30FB      JRNC 8ECC     ;
26    8ED1  FE7F      CP 7F         ;* DEL ?
27    8ED3  CABF91    JP Z,91BF     ;
28    8ED6  FE80      CP 80         ;* CARACTERE REDEFINI (128-138) ?
29    8ED8  3809      JRC 8EE3      ;
30    8EDA  FE8B      CP 8B         ;
31    8EDC  3005      JRNC 8EE3     ;
32    8EDE  C674      ADD A,74      ;oui, alors + 116
33    8EE0  C33692    JP 9236       ;
34    8EE3  FE20      CP 20         ;* CARACTERE NORMAL (32-126) ?
35    8EE5  3805      JRC 8EEC      ;
36    8EE7  FE7F      CP 7F         ;
37    8EE9  DA3692    JP C,9236     ;oui
38    8EEC  FE01      CP 01         ;* SHIFT + ENTER ?
39    8EEE  CAEF93    JP Z,93EF     ;oui
40    8EF1  FE0D      CP 0D         ;* ENTER ?
41    8EF3  CAD393    JP Z,93D3     ;oui
42    8EF6  FEF0      CP F0         ;* CURSEUR (240-251) ?
43    8EF8  3805      JRC 8EFF      ;
44    8EFA  FEFC      CP FC         ;
45    8EFC  DA3A8F    JP C,8F3A     ;oui
46    8EFF  FE00      CP 00         ;* ESPACE FORCE ?
47    8F01  2005      JRNZ 8F08     ;
48    8F03  3EF3      LD A,F3       ;oui alors code 243
```

```
49    8F05  C33692    JP 9236        ;
50    8F08  FE1A      CP 1A          ;* CODE DE CONTROLE ?
51    8F0A  2005      JRNZ 8F11      ;
52    8F0C  3EF1      LD A,F1        ;oui alors code 241
53    8F0E  C33692    JP 9236        ;
54    8F11  FEE0      CP E0          ;* COPY ?
55    8F13  CA8D91    JP Z,918D      ;oui
56    8F16  FE10      CP 10          ;* CLR ?
57    8F18  CA1693    JP Z,9316      ;oui
58    8F1B  FE02      CP 02          ;* SHIFT + CLR ?
59    8F1D  CAB393    JP Z,93B3      ;oui
60    8F20  FE8B      CP 8B          ;* MODE MENU ?
61    8F22  C8        RET Z          ;oui, alors retour Basic
62    8F23  C3CC8E    JP 8ECC        ;* RIEN, ALORS INTERROGATION CLAVIER

      ** BRANCHEMENT CURSEUR **

63    8F3A  FEF3      CP F3          ;* CURSEUR DROITE ?
64    8F3C  CA798F    JP Z,8F79      ;oui
65    8F3F  FEF0      CP F0          ;* CURSEUR HAUT ?
66    8F41  CA8D90    JP Z,908D      ;oui
67    8F44  FEF2      CP F2          ;* CURSEUR GAUCHE ?
68    8F46  CA4190    JP Z,9041      ;oui
69    8F49  FEF1      CP F1          ;* CURSEUR BAS ?
70    8F4B  CADA8F    JP Z,8FDA      ;oui
71    8F4E  FEF4      CP F4          ;* SHIFT + CURSEUR HAUT ?
```

```
72    8F50  CA0492   JP Z,9204      ;oui
73    8F53  FEF5     CP F5          ;* SHIFT + CURSEUR BAS ?
74    8F55  CA1B91   JP Z,911B      ;oui
75    8F58  FEF6     CP F6          ;* SHIFT + CURSEUR GAUCHE ?
76    8F5A  CAE891   JP Z,91E8      ;oui
77    8F5D  FEF7     CP F7          ;* SHIFT + CURSEUR DROIT ?
78    8F5F  CAEB90   JP Z,90EB      ;oui
79    8F62  C3CC8E   JP 8ECC       ;* RIEN, ALORS INTERROGATION CLAVIER


      ** CURSEUR DROITE **

80    8F79  AF       XOR A          ;
81    8F7A  32803A   LD (3A80),A    ;placard 14976 a 0
82    8F7D  CD2995   CALL 9529      ;curseur memoire=maxi?
83    8F80  2827     JRZ 8FA9       ;oui
84    8F82  CDBD95   CALL 95BD      ;afficher caractere
85    8F85  CDA195   CALL 95A1      ;colonne=80 ?
86    8F88  2010     JRNZ 8F9A      ;non
87    8F8A  3E01     LD A,01        ;oui, alors 1 dans placard 14976
88    8F8C  32803A   LD (3A80),A    ;
89    8F8F  CD9B95   CALL 959B      ;ligne ecran=23?
90    8F92  2006     JRNZ 8F9A      ;non
91    8F94  CDD195   CALL 95D1      ;oui,alors scrolling haut
92    8F97  CD2290   CALL 9022      ;et afficher nlle ligne 23
93    8F9A  CDB28F   CALL 8FB2      ;modifier variables
94    8F9D  CDCA95   CALL 95CA      ;afficher curseur
```

```
95    8FA0  3A803A    LD A,(3A80)   ;
96    8FA3  CDFC95    CALL 95FC     ;afficher colonne et ligne theorique
97    8FA6  C3CC8E    JP 8ECC       ;* VERS SAISIE CLAVIER
98    8FA9  CDBD95    CALL 95BD     ;afficher caractere
99    8FAC  CDCA95    CALL 95CA     ;afficher curseur
100   8FAF  C34394    JP 9443       ;coup de cloche
      ** SOUS-PROGRAMME 1 **

101   8FB2  E5        PUSH HL       ;preserver HL
102   8FB3  CD3495    CALL 9534     ;curseur memoire=fin de texte ?
103   8FB6  2005      JRNZ 8FBD     ;non
104   8FB8  3E01      LD A,01       ;oui, alors fin de texte+1
105   8FBA  CD6F95    CALL 956F     ;
106   8FBD  3E01      LD A,01       ;curseur memoire+1
107   8FBF  CD4195    CALL 9541     ;
108   8FC2  21863A    LD HL,3A86    ;HL pointe sur plcrd colonne
109   8FC5  CDA195    CALL 95A1     ;colonne =80 ?
110   8FC8  2803      JRZ 8FCD      ;oui
111   8FCA  34        INC (HL)      ;non, alors colonne+1
112   8FCB  E1        POP HL        ;recuperer HL
113   8FCC  C9        RET           ;
114   8FCD  3601      LD (HL),01    ;colonne=1
115   8FCF  CD9B95    CALL 959B     ;ligne ecran=23 ?
116   8FD2  2802      JRZ 8FD6      ;oui, alors uniquement ligne theorique+1
117   8FD4  2B        DEC HL        ;HL pointe sur plcrd ligne theorique
118   8FD5  34        INC (HL)      ;ligne theorique+1
119   8FD6  2B        DEC HL        ;HL pointe plcrd ligne ecran ou theorique
```

```
120   8FD7  34        INC (HL)     ;ligne ecran ou theorique +1
121   8FD8  E1        POP HL       ;recuperer HL
122   8FD9  C9        RET          ;

      ** CURSEUR BAS **

123   8FDA  CD8F95    CALL 958F    ;ligne theorique=255 ?
124   8FDD  CA4394    JP Z,9443    ;oui, alors coup de cloche
125   8FE0  CDBD95    CALL 95BD    ;afficher caractere
126   8FE3  CD9B95    CALL 959B    ;ligne ecran=23 ?
127   8FE6  2006      JRNZ 8FEE    ;non
128   8FE8  CDD195    CALL 95D1    ;oui, alors scrolling haut...
129   8FEB  CD2290    CALL 9022    ;...et afficher nouvelle ligne 23
130   8FEE  CDFC8F    CALL 8FFC    ;modifier variables
131   8FF1  CDCA95    CALL 95CA    ;afficher curseur
132   8FF4  3E02      LD A,02      ;preparer A pour sp 36
133   8FF6  CDFC95    CALL 95FC    ;sp 36:afficher nlle ligne et nlle colonne
134   8FF9  C3CC8E    JP 8ECC      ;* VERS INTERROGATION CLAVIER

      ** SOUS-PROGRAMME 2 **

135   8FFC  CD3495    CALL 9534    ;fin de texte=curseur memoire ?
136   8FFF  3E50      LD A,50      ;
137   9001  CD4195    CALL 9541    ;curseur memoire+80
138   9004  280B      JRZ 9011     ;oui, fin de texte=curseur memoire
```

```
139   9006  CD3495    CALL 9534     ;non, alors f.d.txt.+80 < cursr. memoire ?
140   9009  300B      JRNC 9016     ;non
141   900B  ED538A3A  LD (3A8A),DE ;oui, alors f.d.txt.=cursr. memoire
142   900F  1805      JR 9016       ;
143   9011  3E50      LD A,50       ;
144   9013  CD6F95    CALL 956F     ;fin de texte+80
145   9016  21843A    LD HL,3A84    ;HL pointe sur ligne ecran
146   9019  CD9B95    CALL 959B     ;ligne ecran=23 ?
147   901C  2801      JRZ 901F      ;oui, alors seulement ligne theorique+1
148   901E  34        INC (HL)      ;sinon, egalement ligne ecran
149   901F  23        INC HL        ;
150   9020  34        INC (HL)      ;
151   9021  C9        RET           ;
      ** SOUS-PROGRAMME 3 **

152   9022  AF        XOR A         ;enlever carry
153   9023  ED5B863A  LD DE,(3A86) ;colonne dans E
154   9027  1600      LD D,00       ;
155   9029  215100    LD HL,0051    ;81 dans HL
156   902C  ED52      SBC HL,DE     ;81-colonne dans HL
157   902E  ED5B883A  LD DE,(3A88) ;curseur memoire dans DE
158   9032  19        ADD HL,DE     ;curseur memoire+81-colonne dans HL
159   9033  115000    LD DE,0050    ;80 dans DE
160   9036  01E0C6    LD BC,C6E0    ;adrsse ecran debut ligne 23 dans HL
161   9039  CDC194    CALL 94C1     ;afficher DE caracteres
162   903C  C9        RET           ;
```

```
      ** CURSEUR GAUCHE **

163   9041  AF        XOR A          ;
164   9042  32803A    LD (3A80),A    ;0 dans placard 14976
165   9045  CD1F95    CALL 951F      ;curseur memoire=mini ?
166   9048  CA4394    JP Z,9443      ;oui, vers coup de cloche
167   904B  CDBD95    CALL 95BD      ;afficher caractere (hl)
168   904E  CDA795    CALL 95A7      ;colonne=1 ?
169   9051  280F      JRZ 9062       ;oui
170   9053  CD7490    CALL 9074      ;modifier variables
171   9056  CDCA95    CALL 95CA      ;afficher curseur
172   9059  3A803A    LD A,(3A80)    ;A pret pour sp 36
173   905C  CDFC95    CALL 95FC      ;sp 36: afficher nlle ligne et nlle colonne
174   905F  C3CC8E    JP 8ECC        ;* VERS INTERROGATION CLAVIER
175   9062  3E01      LD A,01        ;
176   9064  32803A    LD (3A80),A    ;1 dans placard 14976
177   9067  CD9595    CALL 9595      ;ligne ecran=2 ?
178   906A  20E7      JRNZ 9053      ;non
179   906C  CDDF95    CALL 95DF      ;oui, alors scrolling bas
180   906F  CDC790    CALL 90C7      ;afficher nouvelle ligne 2
181   9072  18DF      JR 9053        ;

      ** SOUS-PROGRAMME 4 **

182   9074  3E01      LD A,01        ;
183   9076  CD5095    CALL 9550      ;curseur memoire-1
```

```
184   9079  CDA795    CALL 95A7     ;colonne=1 ?
185   907C  21863A    LD HL,3A86    ;HL pointe sur colonne
186   907F  200A      JRNZ 908B     ;colonne <> 1
187   9081  3650      LD (HL),50    ;oui, colonne=1 alors colonne=80
188   9083  CD9595    CALL 9595     ;ligne ecran=2 ?
189   9086  2802      JRZ 908A      ;oui, alors simplement ligne theorique-1
190   9088  2B        DEC HL        ;sinon, ligne theorique et ligne ecran-1
191   9089  35        DEC (HL)      ;
192   908A  2B        DEC HL        ;
193   908B  35        DEC (HL)      ;
194   908C  C9        RET           ;
```

** CURSEUR HAUT **

```
195   908D  3A853A    LD A,(3A85)   ;
196   9090  FE01      CP 01         ;ligne theorique=1 ?
197   9092  CA4394    JP Z,9443     ;oui, vers coup de cloche
198   9095  CDBD95    CALL 95BD     ;afficher caractere (HL)
199   9098  3E02      LD A,02       ;
200   909A  32803A    LD (3A80),A   ;2 dans placard 14976
201   909D  CD9595    CALL 9595     ;ligne ecran=2 ?
202   90A0  2006      JRNZ 90A8     ;non
203   90A2  CDDF95    CALL 95DF     ;oui, alors scrolling bas
204   90A5  CDC790    CALL 90C7     ;et afficher nouvelle ligne 2
205   90A8  CDB790    CALL 90B7     ;modifier variables
206   90AB  CDCA95    CALL 95CA     ;afficher curseur
207   90AE  3A803A    LD A,(3A80)   ;A pret pour sp 36
```

```
208   90B1  CDFC95   CALL 95FC     ;sp 36: afficher nlle ligne et nlle colonne
209   90B4  C3CC8E   JP 8ECC       ;* VERS INTERROGATION CLAVIER

      ** SOUS-PROGRAMME 5 **

210   90B7  3E50     LD A,50       ;
211   90B9  CD5095   CALL 9550     ;curseur memoire-80
212   90BC  21853A   LD HL,3A85    ;HL pointe sur ligne theorique
213   90BF  35       DEC (HL)      ;ligne theorique-1
214   90C0  CD9595   CALL 9595     ;ligne ecran=2 ?
215   90C3  C8       RET Z         ;oui, fini
216   90C4  2B       DEC HL        ;non, alors ligne ecran-1
217   90C5  35       DEC (HL)      ;
218   90C6  C9       RET           ;

      ** SOUS-PROGRAMME 6 **

219   90C7  2A863A   LD HL,(3A86)  ;
220   90CA  2600     LD H,00       ;colonne dans L
221   90CC  114F00   LD DE,004F    ;79 dans DE
222   90CF  19       ADD HL,DE     ;79+colonne dans HL
223   90D0  EB       EX DE,HL      ;puis dans DE
224   90D1  2A883A   LD HL,(3A88)  ;curseur memoire dans HL
225   90D4  AF       XOR A         ;enlever carry
226   90D5  ED52     SBC HL,DE     ;curseur memoire-DE dans HL
```

```
227   90D7  115000    LD DE,0050    ;80 dans DE
228   90DA  0150C0    LD BC,C050    ;adresse debut ligne 2 dans BC
229   90DD  CDC194    CALL 94C1     ;afficher DE caracteres
230   90E0  C9        RET           ;

      ** SHIFT + CURSEUR DROITE **

231   90EB  CDA195    CALL 95A1     ;colonne=80 ?
232   90EE  CA4394    JP Z,9443     ;oui, vers coup de cloche
233   90F1  CD3495    CALL 9534     ;curseur memoire=fin de texte ?
234   90F4  CA4394    JP Z,9443     ;oui, vers coup de cloche
235   90F7  2A883A    LD HL,(3A88)  ;curseur memoire dans HL
236   90FA  CDBD95    CALL 95BD     ;afficher caractere (HL)
237   90FD  3A863A    LD A,(3A86)   ;colonne dans A
238   9100  47        LD B,A        ;et dans B
239   9101  23        INC HL        ;curseur memoire+1
240   9102  04        INC B         ;colonne+1
241   9103  22883A    LD (3A88),HL  ;ranger curseur memoire+1
242   9106  CD3495    CALL 9534     ;curseur memoire=fin de texte ?
243   9109  78        LD A,B        ;colonne+1 dans A
244   910A  2804      JRZ 9110      ;oui, curseur memoire=fin de texte
245   910C  FE50      CP 50         ;colonne+1=80 ?
246   910E  20F1      JRNZ 9101     ;non, alors boucler
247   9110  3D        DEC A         ;colonne definitive dans A
248   9111  2B        DEC HL        ;curseur memoire definitif dans HL
249   9112  32863A    LD (3A86),A   ;et ranger les deux
250   9115  22883A    LD (3A88),HL  ;
```

```
251   9118  C3798F   JP 8F79        ;* VERS INTERROGATION CLAVIER

      ** SHIFT + CURSEUR BAS **

252   911B  CD3495   CALL 9534      ;curseur memoire=fin de texte ?
253   911E  CA4394   JP Z,9443      ;oui, vers coup de cloche
254   9121  ED5B8A3A LD DE,(3A8A)  ;fin de texte dans DE
255   9125  21983A   LD HL,3A98    ;debut texte dans HL
256   9128  3E01     LD A,01       ;correspond a 1ere ligne theorique
257   912A  01E006   LD BC,06E0    ;nombre de caracteres par page dans BC
258   912D  09       ADD HL,BC     ;HL pointe sur debut page suivante
259   912E  F5       PUSH AF       ;preserver A
260   912F  CDED95   CALL 95ED     ;comparer HL et DE
261   9132  FE01     CP 01         ;HL > DE ?
262   9134  2805     JRZ 913B      ;oui
263   9136  F1       POP AF        ;non, recuperer A
264   9137  C616     ADD A,16      ;ligne theorique+22
265   9139  18F2     JR 912D       ;et boucler
266   913B  ED42     SBC HL,BC     ;revenir a debut page precedente
267   913D  F1       POP AF        ;recuperer ligne theorique correspondante
268   913E  CD5591   CALL 9155     ;sp 7
269   9141  11E006   LD DE,06E0    ;afficher 1760 caracteres
270   9144  0150C0   LD BC,C050    ;a partir adresse depart ligne 2
271   9147  CDC194   CALL 94C1     ;
272   914A  CDCA95   CALL 95CA     ;afficher curseur
273   914D  3E01     LD A,01       ;A pret pour sp 36
```

```
274   914F  CDFC95   CALL 95FC     ;sp 36: afficher nlle ligne et nlle colonne
275   9152  C3CC8E   JP 8ECC       ;* VERS INTERROGATION CLAVIER

      ** SOUS-PROGRAMME 7 **

276   9155  E5       PUSH HL       ;preserver registres
277   9156  F5       PUSH AF       ;
278   9157  EB       EX DE,HL      ;adresse debut de page dans DE
279   9158  2A8A3A   LD HL,(3A8A) ;fin de texte dans HL
280   915B  ED52     SBC HL,DE     ;fin de texte-adresse debut de page dans HL
281   915D  015000   LD BC,0050    ;80 dans DE
282   9160  CD7C91   CALL 917C     ;HL/BC, resultat dans A, reste dans L
283   9163  47       LD B,A        ;resultat dans B
284   9164  F1       POP AF        ;recuperer ligne theorique
285   9165  80       ADD A,B       ;+B
286   9166  32853A   LD (3A85),A   ;ranger nouvelle ligne theorique
287   9169  78       LD A,B        ;calculer ligne ecran (=ligne
288   916A  3C       INC A         ;theorique+2), et ranger
289   916B  3C       INC A         ;dans son placard
290   916C  32843A   LD (3A84),A   ;
291   916F  2C       INC L         ;nouvelle colonne=reste+1
292   9170  7D       LD A,L        ;
293   9171  32863A   LD (3A86),A   ;ranger
294   9174  2A8A3A   LD HL,(3A8A) ;fin de texte dans HL
295   9177  22883A   LD (3A88),HL ;curseur memoire=fin de texte
296   917A  E1       POP HL        ;recuperer HL
297   917B  C9       RET           ;
```

```
      ** SOUS-PROGRAMME 8 **

298   917C  AF        XOR A          ;enlever carry
299   917D  ED42      SBC HL,BC      ;
300   917F  3803      JRC 9184       ;report ?
301   9181  3C        INC A          ;
302   9182  18F9      JR 917D        ;non, continuer boucle
303   9184  ED4A      ADC HL,BC      ;
304   9186  2B        DEC HL         ;reste dans HL
305   9187  C9        RET            ;


      ** METTRE/ENLEVER MODE INSERTION (COPY) **

306   918D  3A873A    LD A,(3A87)    ;semaphore insertion dans A
307   9190  FE01      CP 01          ;deja a 1 ?
308   9192  280A      JRZ 919E       ;oui
309   9194  3E01      LD A,01        ;non, alors mettre semaphore a 1
310   9196  32873A    LD (3A87),A    ;
311   9199  21B991    LD HL,91B9     ;HL pointe sur chaine 'INS'
312   919C  1807      JR 91A5        ;vers affichage
313   919E  AF        XOR A          ;mettre semaphore a 0
314   919F  32873A    LD (3A87),A    ;
315   91A2  21BC91    LD HL,91BC     ;HL pointe sur chaine espaces blancs
316   91A5  E5        PUSH HL        ;* AFFICHAGE
317   91A6  21190E    LD HL,0E19     ;
```

```
318   91A9  CD75BB   CALL BB75     ;positionner curseur en (H,L)
319   91AC  E1       POP HL        ;
320   91AD  0603     LD B,03       ;3 caracteres dans la chaine
321   91AF  7E       LD A,(HL)     ;boucle d'affichage
322   91B0  CD5ABB   CALL BB5A     ;
323   91B3  23       INC HL        ;
324   91B4  10F9     DJNZ 91AF     ;
325   91B6  C3CC8E   JP 8ECC       ;* VERS INTERROGATION CLAVIER
326   91B9  49                     ;'I'
327   91BA  4E                     ;'N'
328   91BB  53                     ;'S'
329   91BC  20                     ;' '
330   91BD  20                     ;' '
331   91BE  20                     ;' '

      ** DEL **

332   91BF  CD1F95   CALL 951F     ;curseur memoire=mini ?
333   91C2  CA4394   JP Z,9443     ;oui, vers coup de cloche
334   91C5  CD3495   CALL 9534     ;curseur memoire=fin de texte ?
335   91C8  3E01     LD A,01       ;
336   91CA  CC7E95   CALL Z,957E   ;si oui, fin de texte-A (donc -1)
337   91CD  215F90   LD HL,905F    ;* TRANSFORMER CURSEUR GAUCHE EN SP
338   91D0  36C9     LD (HL),C9    ;
339   91D2  E5       PUSH HL       ;
340   91D3  CD4190   CALL 9041     ;appeller CURSEUR GAUCHE comme un sp
```

```
341   91D6  E1        POP HL         ;* RESTAURER CURSEUR GAUCHE
342   91D7  36C3      LD (HL),C3     ;
343   91D9  2A883A    LD HL,(3A88)   ;
344   91DC  36FF      LD (HL),FF     ;code espace inutile dans (HL)
345   91DE  CDBD95    CALL 95BD      ;afficher caractere (HL)
346   91E1  CDCA95    CALL 95CA      ;afficher curseur
347   91E4  C3CC8E    JP 8ECC        ;* VERS INTERROGATION CLAVIER
348  91E7  00        NOP            ;
     ** SHIFT + CURSEUR GAUCHE **

349   91E8  CDA795    CALL 95A7      ;colonne=1 ?
350   91EB  CA4394    JP Z,9443      ;oui, vers coup de cloche
351   91EE  CDBD95    CALL 95BD      ;afficher caractere (HL)
352   91F1  3A863A    LD A,(3A86)    ;colonne dans A
353   91F4  3D        DEC A          ;-2
354   91F5  3D        DEC A          ;`
355   91F6  FE00      CP 00          ;colonne-2 = 0 ?
356   91F8  CA4190    JP Z,9041      ;oui, alors vers CURSEUR GAUCHE
357   91FB  47        LD B,A         ;compteur de boucle
358   91FC  CD7490    CALL 9074      ;modifier variables comme pour cursr gauche
359   91FF  10FB      DJNZ 91FC      ;boucler
360   9201  C34190    JP 9041        ;* VERS CURSEUR GAUCHE


     ** SHIFT + CURSEUR HAUT **

361   9204  CD1F95    CALL 951F      ;curseur memoire=mini ?
```

```
362   9207  CA4394    JP Z,9443     ;oui, vers coup de cloche
363   920A  21843A    LD HL,3A84    ;ligne ecran=2
364   920D  3602      LD (HL),02    ;
365   920F  23        INC HL        ;ligne theorique=1
366   9210  3601      LD (HL),01    ;
367   9212  23        INC HL        ;colonne=1
368   9213  3601      LD (HL),01    ;
369   9215  21983A    LD HL,3A98    ;curseur memoire=15000
370   9218  22883A    LD (3A88),HL  ;
371   921B  117841    LD DE,4178    ;afficher ecran de 15000 a 16760
372   921E  CD8A94    CALL 948A     ;
373   9221  AF        XOR A         ;afficher nouvelle colonne
374   9222  CDFC95    CALL 95FC     ;
375   9225  3E02      LD A,02       ;afficher nouvelle ligne
376   9227  CDFC95    CALL 95FC     ;
377   922A  3E20      LD A,20       ;effacer eventuel 3eme chiffre ancien. lig.
378   922C  CD5ABB    CALL BB5A     ;
379   922F  CDCA95    CALL 95CA     ;afficher curseur
380   9232  C3CC8E    JP 8ECC       ;* VERS INTERROGATION CLAVIER


      ** SAISIE D'UN CARACTERE **

381   9236  F5        PUSH AF       ;preserver caractere
382   9237  3A873A    LD A,(3A87)   ;semaphore d'insertion=1 ?
383   923A  FE01      CP 01         ;
384   923C  2004      JRNZ 9242     ;non
```

```
385   923E  F1       POP AF         ;oui, recuperer caractere...
386   923F  C34A92   JP 924A        ;... et vers insertion
387   9242  F1       POP AF         ;recuperer caractere
388   9243  2A883A   LD HL,(3A88)   ;ranger en RAM a emplacement
389   9246  77       LD (HL),A      ;pointe par curseur memoire
390   9247  C3798F   JP 8F79        ;* VERS CURSEUR DROITE
      ** INSERTION D'UN CARACTERE **

391   924A  32813A   LD (3A81),A    ;rangement provisoire du caractere
392   924D  CD2995   CALL 9529      ;curseur memoire=maxi ?
393   9250  CA4394   JP Z,9443      ;oui, vers coup de cloche
394   9253  EB       EX DE,HL       ;curseur memoire dans HL
395   9254  3EFF     LD A,FF        ;cursr. mem. pointe sur espace inutile ?
396   9256  BE       CP (HL)        ;
397   9257  3A813A   LD A,(3A81)    ;recuperer caracter
398   925A  CA4392   JP Z,9243      ;* OUI, VERS FIN PROGRAMME PRECEDENT
399   925D  CDCB92   CALL 92CB      ;f.d.t. ou esp. inut. ou fin paragr. ds DE
400   9260  FE03     CP 03          ;fin de texte ?
401   9262  2019     JRNZ 927D      ;non
402   9264  CD6195   CALL 9561      ;oui, fin de texte=maxi ?
403   9267  3E01     LD A,01        ;
404   9269  C46F95   CALL NZ,956F  ;non, alors fin de texte+1
405   926C  1B       DEC DE         ;DE pointe sur caract. avant espace inutile
406   926D  3E01     LD A,01        ;decaler (HL) a (DE) de +1
407   926F  CD5594   CALL 9455      ;
408   9272  3A813A   LD A,(3A81)    ;recuperer caractere
409   9275  77       LD (HL),A      ;et l'ecrire en RAM
```

```
410   9276  13         INC DE          ;
411   9277  CD8A94     CALL 948A       ;afficher de (HL) a (DE)
412   927A  C3798F     JP 8F79         ;* VERS CURSEUR DROITE
413   927D  FE01       CP 01           ;espace inutile ?
414   927F  28EB       JRZ 926C        ;oui
415   9281  3E50       LD A,50         ;non, donc fin paragr. / colonne=80 ?
416   9283  B8         CP B            ;
417   9284  20E7       JRNZ 926D       ;non
418   9286  E5         PUSH HL         ;oui, preserver HL et DE
419   9287  D5         PUSH DE         ;
420   9288  13         INC DE          ;pointer HL sur DE+1 et DE sur f.d.t
421   9289  EB         EX DE,HL        ;
422   928A  ED5B8A3A   LD DE,(3A8A)    ;
423   928E  3E50       LD A,50         ;decaler (HL) a (DE) de +80
424   9290  CD5594     CALL 9455       ;
425   9293  D1         POP DE          ;recuperer DE et HL
426   9294  E1         POP HL          ;
427   9295  3E01       LD A,01         ;decaler (HL) a (DE) de +1
428   9297  CD5594     CALL 9455       ;
429   929A  13         INC DE          ;ecrire 79 espaces inutiles a partir
430   929B  13         INC DE          ;de fin de paragraphe +2
431   929C  064F       LD B,4F         ;
432   929E  CDFE92     CALL 92FE       ;
433   92A1  3E50       LD A,50         ;fin de texte+80
434   92A3  CD6F95     CALL 956F       ;
435   92A6  E5         PUSH HL         ;preserver HL
436   92A7  ED5B8A3A   LD DE,(3A8A)    ;fin de texte+80 > 35399 ?
```

```
437   92AB  21478A    LD HL,8A47    ;
438   92AE  CDED95    CALL 95ED     ;
439   92B1  FEFF      CP FF         ;
440   92B3  2006      JRNZ 92BB     ;non
441   92B5  228A3A    LD (3A8A),HL  ;oui, alors fin de texte=35399...
442   92B8  CD0793    CALL 9307     ;...et nettoyer 35400 a 35480
443   92BB  E1        POP HL        ;recuperer HL
444   92BC  3A813A    LD A,(3A81)   ;recuperer caractere...
445   92BF  77        LD (HL),A     ;...et l'ecrire en RAM
446   92C0  CD8A94    CALL 948A     ;afficher caracteres de (HL) a (DE)
447   92C3  C3798F    JP 8F79       ;* VERS CURSEUR DROITE
      ** SOUS-PROGRAMME 9 **

448   92CB  3A863A    LD A,(3A86)   ;colonne dans B
449   92CE  47        LD B,A        ;
450   92CF  ED5B8A3A  LD DE,(3A8A)  ;fin de texte dans DE
451   92D3  E5        PUSH HL       ;preserver HL
452   92D4  CDED95    CALL 95ED     ;comparer HL et DE
453   92D7  FE00      CP 00         ;HL=DE ?
454   92D9  2004      JRNZ 92DF     ;non
455   92DB  E1        POP HL        ;oui, alors f.d.t. trouve/Recuperer HL
456   92DC  3E03      LD A,03       ;signal dans A
457   92DE  C9        RET           ;fini
458   92DF  7E        LD A,(HL)     ;espace inutile ?
459   92E0  FEFF      CP FF         ;
460   92E2  2005      JRNZ 92E9     ;non
461   92E4  EB        EX DE,HL      ;oui, alors adresse correspondante dans DE
```

```
462   92E5  E1        POP HL          ;recuperer HL
463   92E6  3E01      LD A,01         ;signal dans A
464   92E8  C9        RET             ;fini
465   92E9  7E        LD A,(HL)       ;fin de paragraphe ?
466   92EA  FEF2      CP F2           ;
467   92EC  2005      JRNZ 92F3       ;non
468   92EE  EB        EX DE,HL        ;oui, alors adresse correspondante dans DE
469   92EF  E1        POP HL          ;recuperer HL
470   92F0  3E02      LD A,02         ;signal dans A
471   92F2  C9        RET             ;fini
472   92F3  04        INC B           ;rien trouve, alors B+1 (colonne)
473   92F4  23        INC HL          ;HL pointe sur adresse suivante
474   92F5  3E51      LD A,51         ;B=81 ?
475   92F7  B8        CP B            ;
476   92F8  20DA      JRNZ 92D4       ;non, alors continuer a chercher
477   92FA  0601      LD B,01         ;oui, alors B=1
478   92FC  18D6      JR 92D4         ;continuer a chercher

      ** SOUS-PROGRAMME 10 **

479   92FE  D5        PUSH DE         ;preserver DE
480   92FF  3EFF      LD A,FF         ;code espace inutile dans A
481   9301  12        LD (DE),A       ;ecrire en RAM
482   9302  13        INC DE          ;adrsse suivante
483   9303  10FC      DJNZ 9301       ;si compteur <> 0, continuer
484   9305  D1        POP DE          ;recuperer DE
```

```
485   9306  C9        RET             ;

      ** SOUS-PROGRAMME 11 **

486   9307  E5        PUSH HL         ;preserver registres
487   9308  C5        PUSH BC         ;
488   9309  21488A    LD HL,8A48      ;pointer HL sur 35400
489   930C  0650      LD B,50         ;compteur
490   930E  36FF      LD (HL),FF      ;ecrire espace inutile en RAM
491   9310  23        INC HL          ;adresse suivante
492   9311  10FB      DJNZ 930E       ;si compteur <> 0, continuer
493   9313  C1        POP BC          ;recuperer registres
494   9314  E1        POP HL          ;
495   9315  C9        RET             ;
      ** CLR **

496   9316  CD2995    CALL 9529       ;curseur memoire=maxi ?
497   9319  EB        EX DE,HL        ;curseur memoire dans HL
498   931A  200B      JRNZ 9327       ;curseur memoire < maxi
499   931C  36FF      LD (HL),FF      ;code espace inutile en RAM
500   931E  CDBD95    CALL 95BD       ;afficher caractere (HL)
501   9321  CDCA95    CALL 95CA       ;afficher curseur
502   9324  C3CC8E    JP 8ECC         ;* VERS INTERROGATION CLAVIER
503   9327  CD3495    CALL 9534       ;curseur memoire=fin de texte ?
504   932A  CA4394    JP Z,9443       ;oui, vers coup de cloche
505   932D  21E292    LD HL,92E2      ;supprimer, dans sp 9, partie concernant...
```

```
506   9330  3618      LD (HL),18    ;...espace inutile
507   9332  EB        EX DE,HL      ;curseur memoire dans HL
508   9333  CDCB92    CALL 92CB     ;adresse fin paragr. ou esp. inut. dans DE
509   9336  E5        PUSH HL       ;preserver HL
510   9337  21E292    LD HL,92E2    ;restaurer sp 9
511   933A  3620      LD (HL),20    ;
512   933C  E1        POP HL        ;recuperer HL
513   933D  FE02      CP 02         ;fin de paragraphe ?
514   933F  2815      JRZ 9356      ;oui
515   9341  3E01      LD A,01       ;non, fin de texte-1
516   9343  CD7E95    CALL 957E     ;
517   9346  23        INC HL        ;decaler zone (HL+1) a (DE) de -1
518   9347  3E01      LD A,01       ;
519   9349  CD6F94    CALL 946F     ;
520   934C  2B        DEC HL        ;afficher zone (HL) a (DE)
521   934D  CD8A94    CALL 948A     ;
522   9350  CDCA95    CALL 95CA     ;afficher curseur
523   9353  C3CC8E    JP 8ECC       ;* VERS INTERROGATION CLAVIER
524   9356  CDED95    CALL 95ED     ;fin de paragraphe / comparer HL et DE
525   9359  FE00      CP 00         ;HL=DE ?
526   935B  2845      JRZ 93A2      ;oui
527   935D  C5        PUSH BC       ;non, preserver colonne
528   935E  23        INC HL        ;decaler zone (HL+1) a DE de -1
529   935F  3E01      LD A,01       ;
530   9361  CD6F94    CALL 946F     ;
531   9364  2B        DEC HL        ;restaurer HL
532   9365  3EFF      LD A,FF       ;effacer anciennne fin de paragraphe
```

```
533   9367  12        LD (DE),A     ;
534   9368  C1        POP BC        ;recuperer colonne
535   9369  3E01      LD A,01       ;colonne=1 ?
536   936B  B8        CP B          ;
537   936C  20DF      JRNZ 934D     ;non
538   936E  E5        PUSH HL       ;oui, preserver HL
539   936F  2A8A3A    LD HL,(3A8A)  ;fin de texte dans HL
540   9372  E5        PUSH HL       ;ranger
541   9373  EB        EX DE,HL      ;fin de texte dans DE
542   9374  015000    LD BC,0050    ;fin de texte+80 dans HL
543   9377  09        ADD HL,BC     ;
544   9378  CDED95    CALL 95ED     ;comparer HL et DE
545   937B  FE01      CP 01         ;HL > DE ?
546   937D  2007      JRNZ 9386     ;non
547   937F  D1        POP DE        ;oui, nettoyer pile
548   9380  E5        PUSH HL       ;fin de texte+80 sur pile...
549   9381  E5        PUSH HL       ;... dans DE...
550   9382  D1        POP DE        ;
551   9383  228A3A    LD (3A8A),HL  ;... et dans placard
552   9386  3E50      LD A,50       ;decaler zone (HL) a (DE) de -80
553   9388  CD6F94    CALL 946F     ;
554   938B  3E50      LD A,50       ;fin de texte-80
555   938D  CD7E95    CALL 957E     ;
556   9390  ED5B8A3A  LD DE,(3A8A)  ;nettoyer zone f.d.t. a f.d.t.+80
557   9394  CDFE92    CALL 92FE     ;
558   9397  D1        POP DE        ;recuperer registres
559   9398  E1        POP HL        ;
```

```
560   9399  CD8A94   CALL 948A     ;afficher de (HL) a (DE)
561   939C  CDCA95   CALL 95CA     ;afficher curseur
562   939F  C3CC8E   JP 8ECC       ;* VERS INTERROGATION CLAVIER
563   93A2  3E01     LD A,01       ;colonne=1 ?
564   93A4  B8       CP B          ;
565   93A5  28C7     JRZ 936E      ;oui
566   93A7  36FF     LD (HL),FF    ;non, effacer caractere...
567   93A9  CDBD95   CALL 95BD     ;
568   93AC  18EE     JR 939C       ;... et decalage RAM

      ** SHIFT + CLR **

569   93B3  CD3495   CALL 9534     ;curseur memoire=fin de texte ?
570   93B6  CA4394   JP Z,9443     ;oui, vers coup de cloche
571   93B9  CDA795   CALL 95A7     ;colonne=1 ?
572   93BC  280D     JRZ 93CB      ;oui
573   93BE  215F90   LD HL,905F    ;non, forcer cursr. en col.1:transformer...
574   93C1  36C9     LD (HL),C9    ;...SHIFT+GAUCHE en sp...
575   93C3  CDE891   CALL 91E8     ;... et appeler
576   93C6  215F90   LD HL,905F    ;restaurer SHIFT+GAUCHE
577   93C9  36C3     LD (HL),C3    ;
578   93CB  2A883A   LD HL,(3A88)  ;code fin paragr. a emplac. cursr. mem.
579   93CE  36F2     LD (HL),F2    ;
580   93D0  C31693   JP 9316       ;* VERS CLR
```

```
      ** ENTER **

581   93D3  CD8F95   CALL 958F     ;ligne actuelle=maxi ?
582   93D6  CA4394   JP Z,9443     ;oui, vers coup de cloche
583   93D9  CDA795   CALL 95A7     ;colonne actuelle=1 ?
584   93DC  CADA8F   JP Z,8FDA     ;oui, alors simplement vers CRSR BAS
585   93DF  215F90   LD HL,905F   ;non, forcer crsr. en col. 1:transformer...
586   93E2  36C9     LD (HL),C9    ;...SHIFT+CURSEUR GAUCHE en sp...
587   93E4  CDE891   CALL 91E8     ;... et appeler
588   93E7  215F90   LD HL,905F   ;restaurer SHIFT+CURSEUR GAUCHE
589   93EA  36C3     LD (HL),C3    ;
590   93EC  C3DA8F   JP 8FDA       ;* VERS CURSEUR BAS

      ** SHIFT + ENTER **

591   93EF  CD8F95   CALL 958F     ;ligne theorique=maxi ?
592   93F2  CA4394   JP Z,9443     ;oui, vers coup de cloche
593   93F5  CD3495   CALL 9534     ;curseur memoire=fin de texte ?
594   93F8  202A     JRNZ 9424     ;non, traiter en insertion
595   93FA  CDA195   CALL 95A1     ;colonne=80 ?
596   93FD  281D     JRZ 941C      ;oui, sortir code fin de par. normalement
597   93FF  2A883A   LD HL,(3A88)  ;non, ecrire code en RAM
598   9402  36F2     LD (HL),F2    ;
599   9404  CDBD95   CALL 95BD     ;et afficher
600   9407  3A863A   LD A,(3A86)   ;colonne dans A...
601   940A  47       LD B,A        ;...puis dans B
602   940B  3E50     LD A,50       ;
```

```
603   940D  90        SUB B          ;80-colonne dans A
604   940E  CD6F95    CALL 956F      ;fin de texte+A
605   9411  CD4195    CALL 9541      ;curseur memoire+A
606   9414  3E50      LD A,50        ;colonne=80
607   9416  32863A    LD (3A86),A    ;
608   9419  C3798F    JP 8F79        ;* VERS CURSEUR DROITE
609   941C  2A883A    LD HL,(3A88)   ;code fin de paragraphe en RAM
610   941F  36F2      LD (HL),F2     ;
611   9421  C3798F    JP 8F79        ;* VERS CURSEUR DROITE
612   9424  21A68F    LD HL,8FA6     ;transformer CURSEUR DROITE en sp
613   9427  36C9      LD (HL),C9     ;
614   9429  21863A    LD HL,3A86     ;* CALCULER 81-COLONNE
615   942C  3E51      LD A,51        ;
616   942E  96        SUB (HL)       ;
617   942F  47        LD B,A         ;resultat dans B
618   9430  3EF2      LD A,F2        ;* INSERER 1 FOIS CODE FIN DE PARAGRAPHE
619   9432  C5        PUSH BC        ;ET B-1 FOIS CODE ESPACE INUTILE
620   9433  CD4A92    CALL 924A      ;
621   9436  C1        POP BC         ;
622   9437  3EFF      LD A,FF        ;
623   9439  10F7      DJNZ 9432      ;
624   943B  21A68F    LD HL,8FA6     ;restaurer CURSEUR DROITE
625   943E  36C3      LD (HL),C3     ;
626   9440  C3CC8E    JP 8ECC        ;* VERS INTERROGATION CLAVIER
```

```
      ** ERREUR **

627   9443  3E07      LD A,07       ;code 7 dans A
628   9445  CD5ABB    CALL BB5A     ;executer
629   9448  C3CC8E    JP 8ECC       ;* VERS INTERROGATION CLAVIER


      ** SOUS-PROGRAMME 12 **

630   9455  47        LD B,A        ;preserver A
631   9456  AF        XOR A         ;enlever Carry
632   9457  78        LD A,B        ;recuperer A
633   9458  E5        PUSH HL       ;preserver registres
634   9459  D5        PUSH DE       ;
635   945A  EB        EX DE,HL      ;* CALCULER NOMBRE D'OCTETS A DECALER...
636   945B  ED52      SBC HL,DE     ;... SOIT: DE-HL+1
637   945D  23        INC HL        ;
638   945E  E5        PUSH HL       ;
639   945F  C1        POP BC        ;resultat dans BC
640   9460  E1        POP HL        ;* CALCULER POSITION DE POUR DECALAGE...
641   9461  E5        PUSH HL       ;... SOIT: DE+A
642   9462  5F        LD E,A        ;
643   9463  1600      LD D,00       ;
644   9465  19        ADD HL,DE     ;
645   9466  E5        PUSH HL       ;
646   9467  D1        POP DE        ;resultat dans DE
647   9468  E1        POP HL        ;recuperer HL initial
```

```
648   9469  E5        PUSH HL       ;et preserver a nouveau
649   946A  EDB8      LDDR          ;transfert repetitif avec decrementation
650   946C  D1        POP DE        ;recuperer registres
651   946D  E1        POP HL        ;
652   946E  C9        RET           ;

      ** SOUS-PROGRAMME 13 **

653   946F  47        LD B,A        ;preserver A
654   9470  AF        XOR A         ;enlever carry
655   9471  78        LD A,B        ;recuperer A
656   9472  D5        PUSH DE       ;preserver registres
657   9473  E5        PUSH HL       ;
658   9474  EB        EX DE,HL      ;* CALCULER NOMBRE D'OCTETS A DECALER...
659   9475  ED52      SBC HL,DE     ;... SOIT: DE-HL+1
660   9477  23        INC HL        ;
661   9478  E5        PUSH HL       ;
662   9479  C1        POP BC        ;resultat dans BC
663   947A  E1        POP HL        ;* CALCULER POSITION DE POUR DECALAGE...
664   947B  E5        PUSH HL       ;... SOIT: DE+A
665   947C  5F        LD E,A        ;
666   947D  1600      LD D,00       ;
667   947F  ED52      SBC HL,DE     ;
668   9481  E5        PUSH HL       ;
669   9482  D1        POP DE        ;resultat dans DE
670   9483  E1        POP HL        ;recuperer HL initial
671   9484  E5        PUSH HL       ;preserver a nouveau
```

```
672   9485  EDB0      LDIR          ;transfert repetitif avec incrementation
673   9487  E1        POP HL        ;recuperer registres
674   9488  D1        POP DE        ;
675   9489  C9        RET           ;

      ** SOUS-PROGRAMME 14 **

676   948A  D5        PUSH DE       ;preserver registres
677   948B  E5        PUSH HL       ;
678   948C  AF        XOR A         ;enlever carry
679   948D  EB        EX DE,HL      ;* CALCULER NOMBRE D'OCTETS A AFFICHER...
680   948E  ED52      SBC HL,DE     ;... SOIT: DE-HL+1
681   9490  23        INC HL        ;
682   9491  EB        EX DE,HL      ;resultat dans DE
683   9492  3A843A    LD A,(3A84)   ;ligne dans L
684   9495  6F        LD L,A        ;
685   9496  3A863A    LD A,(3A86)   ;colonne dans H
686   9499  67        LD H,A        ;
687   949A  25        DEC H         ;colonne reelle = colonne -1
688   949B  2D        DEC L         ;ligne reelle = ligne -1
689   949C  CD1ABC    CALL BC1A     ;calculer adresse ecran correspondante
690   949F  E5        PUSH HL       ;resultat dans HL, preserver
691   94A0  19        ADD HL,DE     ;adresse + nombre d'octets > 65535 ?
692   94A1  3007      JRNC 94AA     ;non
693   94A3  11E006    LD DE,06E0    ;oui, alors nbre octets a afficher=1760
694   94A6  E1        POP HL        ;adresse ecran dans HL
695   94A7  E5        PUSH HL       ;et preserver a nouveau
```

```
696   94A8  19        ADD HL,DE     ;adresse ecran + nombre d'octets dans HL
697   94A9  AF        XOR A         ;enlever carry
698   94AA  012FC7    LD BC,C72F    ;* RESULTAT <=50991 (FIN 23eme LIGNE) ?
699   94AD  ED42      SBC HL,BC     ;
700   94AF  3009      JRNC 94BA     ;non
701   94B1  C1        POP BC        ;oui, alors adresse ecran de depart ds BC
702   94B2  E1        POP HL        ;adresse initiale dans HL
703   94B3  E5        PUSH HL       ;et preserver a nouveau
704   94B4  CDC194    CALL 94C1     ;afficher DE caracteres
705   94B7  E1        POP HL        ;recuperer registres
706   94B8  D1        POP DE        ;
707   94B9  C9        RET           ;fin
708   94BA  2B        DEC HL        ;valeur du depassement dans HL
709   94BB  EB        EX DE,HL      ;puis dans DE, et HL contient DE initial
710   94BC  ED52      SBC HL,DE     ;DE initial - depassement
711   94BE  EB        EX DE,HL      ;
712   94BF  18F0      JR 94B1       ;continuer

      ** SOUS-PROGRAMME 15 **

713   94C1  D5        PUSH DE       ;preserver registres
714   94C2  E5        PUSH HL       ;
715   94C3  7E        LD A,(HL)     ;caractere dans A
716   94C4  CDD194    CALL 94D1     ;afficher caractere
717   94C7  E1        POP HL        ;recuperer registres
718   94C8  D1        POP DE        ;
```

```
719   94C9  1B       DEC DE       ;* COMPTEUR = 0 ?
720   94CA  7B       LD A,E       ;
721   94CB  B2       OR D         ;
722   94CC  C8       RET Z        ;oui, fini
723   94CD  23       INC HL       ;pointeur RAM sur adresse suivante
724   94CE  03       INC BC       ;pointeur ecran sur adresse suivante
725   94CF  18F0     JR 94C1      ;continuer

      ** SOUS-PROGRAMME 16 **

726   94D1  FEEB     CP EB        ;caractere redefini ( >235 ?)
727   94D3  3022     JRNC 94F7    ;oui
728   94D5  5F       LD E,A       ;non, code ASCII dans E
729   94D6  1600     LD D,00      ;annuler D
730   94D8  210000   LD HL,0000   ;annuler HL
731   94DB  3E08     LD A,08      ;compteur dans A
732   94DD  19       ADD HL,DE    ;* BOUCLE DE MULTIPLICATION
733   94DE  3D       DEC A        ;
734   94DF  20FC     JRNZ 94DD    ;si fini, HL=8*code ASCII / sinon continuer
735   94E1  7C       LD A,H       ;ajouter &38 a H, equivaut a ajouter...
736   94E2  C638     ADD A,38     ;...14336 a HL (14336=256*&38)...
737   94E4  67       LD H,A       ;...donc HL=8*code ASCII+14336
738   94E5  50       LD D,B       ;adresse ecran dans DE
739   94E6  59       LD E,C       ;
740   94E7  CD06B9   CALL B906    ;selectionner la ROM
741   94EA  7E       LD A,(HL)    ;charger A avec contenu ROM
```

```
742   94EB  12       LD (DE),A    ;et adresse ecran avec A
743   94EC  23       INC HL       ;HL pointe sur adresse ROM suivante
744   94ED  7A       LD A,D       ;ajouter 2048 a DE, qui pointe donc sur...
745   94EE  C608     ADD A,08     ;...octet ecran situe...
746   94F0  57       LD D,A       ;...immediatement sous le precedent
747   94F1  30F7     JRNC 94EA    ;si pas de depassement, boucler
748   94F3  CD09B9   CALL B909    ;si depassement, couper ROM select
749   94F6  C9       RET          ;et retour
750   94F7  D6EB     SUB EB       ;* CALCULER (A-235)*8+42580
751   94F9  5F       LD E,A       ;
752   94FA  1600     LD D,00      ;
753   94FC  210000   LD HL,0000   ;
754   94FF  3E08     LD A,08      ;
755   9501  19       ADD HL,DE    ;
756   9502  3D       DEC A        ;
757   9503  20FC     JRNZ 9501    ;
758   9505  1154A6   LD DE,A654   ;
759   9508  19       ADD HL,DE    ;resultat dans HL
760   9509  50       LD D,B       ;* BOUCLE DE SORTIE ECRAN IDENTIQUE...
761   950A  59       LD E,C       ;...A CELLE UTILISEE POUR CARACTERES...
762   950B  7E       LD A,(HL)    ;...NORMAUX, MAIS SANS SELECTION...
763   950C  12       LD (DE),A    ;...DE LA ROM
764   950D  23       INC HL       ;
765   950E  7A       LD A,D       ;
766   950F  C608     ADD A,08     ;
767   9511  57       LD D,A       ;
768   9512  30F7     JRNC 950B    ;
```

```
769   9514  C9        RET             ;

      ** SOUS-PROGRAMME 17 **

770   951F  AF        XOR A           ;enlever carry
771   9520  2A883A    LD HL,(3A88)    ;curseur memoire dans HL
772   9523  11983A    LD DE,3A98      ;15000 dans DE
773   9526  ED52      SBC HL,DE       ;HL-DE
774   9528  C9        RET             ;

      ** SOUS-PROGRAMME 18 **

775   9529  AF        XOR A           ;enlever carry
776   952A  21478A    LD HL,8A47      ;35399 dans HL
777   952D  ED5B883A  LD DE,(3A88)    ;curseur memoire dans DE
778   9531  ED52      SBC HL,DE       ;HL-DE
779   9533  C9        RET             ;

      ** SOUS-PROGRAMME 19 **

780   9534  AF        XOR A           ;enlever carry
781   9535  E5        PUSH HL         ;preserver HL
782   9536  2A8A3A    LD HL,(3A8A)    ;fin de texte dans HL
783   9539  ED5B883A  LD DE,(3A88)    ;curseur menoire dans DE
```

```
784   953D  ED52      SBC HL,DE     ;HL-DE
785   953F  E1        POP HL        ;recuperer HL
786   9540  C9        RET           ;

      ** SOUS-PROGRAMME 20 **

787   9541  E5        PUSH HL       ;preserver registres
788   9542  D5        PUSH DE       ;
789   9543  5F        LD E,A        ;valeur a ajouter dans DE
790   9544  1600      LD D,00       ;
791   9546  2A883A    LD HL,(3A88)  ;curseur memoire dans HL
792   9549  19        ADD HL,DE     ;
793   954A  22883A    LD (3A88),HL ;HL+DE dans placard curseur memoire
794   954D  D1        POP DE        ;recuperer registres
795   954E  E1        POP HL        ;
796   954F  C9        RET           ;
      ** SOUS-PROGRAMME 21 **

797   9550  E5        PUSH HL       ;preserver registres
798   9551  D5        PUSH DE       ;
799   9552  5F        LD E,A        ;valeur a soustraire dans DE
800   9553  1600      LD D,00       ;
801   9555  AF        XOR A         ;enlever carry
802   9556  2A883A    LD HL,(3A88)  ;curseur memoire dans HL
803   9559  ED52      SBC HL,DE     ;
804   955B  22883A    LD (3A88),HL ;HL-DE dans placard curseur memoire
```

```
805   955E  D1        POP DE        ;recuperer registres
806   955F  E1        POP HL        ;
807   9560  C9        RET           ;

      ** SOUS-PROGRAMME 22 **

808   9561  AF        XOR A         ;enlever carry
809   9562  E5        PUSH HL       ;preserver registres
810   9563  C5        PUSH BC       ;
811   9564  2A8A3A    LD HL,(3A8A)  ;fin de texte dans HL
812   9567  01478A    LD BC,8A47    ;35399 dans BC
813   956A  ED42      SBC HL,BC     ;
814   956C  C1        POP BC        ;recuperer registres
815   956D  E1        POP HL        ;
816   956E  C9        RET           ;

      ** SOUS-PROGRAMME 23 **

817   956F  E5        PUSH HL       ;preserver registres
818   9570  D5        PUSH DE       ;
819   9571  5F        LD E,A        ;valeur a ajouter dans DE
820   9572  1600      LD D,00       ;
821   9574  2A8A3A    LD HL,(3A8A)  ;fin de texte dans HL
822   9577  19        ADD HL,DE     ;
823   9578  228A3A    LD (3A8A),HL  ;HL+DE dans placard fin de texte
```

```
824   957B  D1        POP DE        ;recuperer registres
825   957C  E1        POP HL        ;
826   957D  C9        RET           ;

      ** SOUS-PROGRAMME 24 **

827   957E  E5        PUSH HL       ;preserver registres
828   957F  D5        PUSH DE       ;
829   9580  5F        LD E,A        ;valeur a soustraire dans DE
830   9581  1600      LD D,00       ;
831   9583  AF        XOR A         ;enlever carry
832   9584  2A8A3A    LD HL,(3A8A)  ;fin de texte dans HL
833   9587  ED52      SBC HL,DE     ;
834   9589  228A3A    LD (3A8A),HL  ;HL-DE dans placard fin de texte
835   958C  D1        POP DE        ;recuperer registres
836   958D  E1        POP HL        ;
837   958E  C9        RET           ;

      ** SOUS-PROGRAMME 25 **

838   958F  3A853A    LD A,(3A85)   ;ligne theorique dans A
839   9592  FEFF      CP FF         ;= 255 ?
840   9594  C9        RET           ;
```

```
      ** SOUS-PROGRAMME 26 **

841   9595  3A843A    LD A,(3A84)  ;ligne ecran dans A
842   9598  FE02      CP 02        ;= 2 ?
843   959A  C9        RET          ;

      ** SOUS-PROGRAMME 27 **

844   959B  3A843A    LD A,(3A84)  ;ligne ecran dans A
845   959E  FE17      CP 17        ;= 23 ?
846   95A0  C9        RET          ;

      ** SOUS-PROGRAMME 28 **

847   95A1  3A863A    LD A,(3A86)  ;colonne dans A
848   95A4  FE50      CP 50        ;= 80 ?
849   95A6  C9        RET          ;

      ** SOUS-PROGRAMME 29 **

850   95A7  3A863A    LD A,(3A86)  ;colonne dans A
851   95AA  FE01      CP 01        ;= 1 ?
852   95AC  C9        RET          ;
```

```
      ** SOUS-PROGRAMME 30 **

853   95AD  E5        PUSH HL       ;preserver registres
854   95AE  F5        PUSH AF       ;
855   95AF  3A843A    LD A,(3A84)   ;ligne ecran dans A
856   95B2  6F        LD L,A        ;puis dans L
857   95B3  3A863A    LD A,(3A86)   ;colonne dans A
858   95B6  67        LD H,A        ;puis dans H
859   95B7  CD75BB    CALL BB75     ;curseur systeme en (H,L)
860   95BA  F1        POP AF        ;recuperer registres
861   95BB  E1        POP HL        ;
862   95BC  C9        RET           ;

      ** SOUS-PROGRAMME 31 **

863   95BD  E5        PUSH HL       ;preserver HL
864   95BE  2A883A    LD HL,(3A88)  ;curseur memoire dans HL
865   95C1  7E        LD A,(HL)     ;caractere pointe par HL dans A
866   95C2  CDAD95    CALL 95AD     ;curseur systeme a position variables
867   95C5  CD5ABB    CALL BB5A     ;afficher caractere
868   95C8  E1        POP HL        ;recuperer HL
869   95C9  C9        RET           ;
```

```
      ** SOUS-PROGRAMME 32 **

870   95CA  CDAD95    CALL 95AD     ;curseur systeme a position variable
871   95CD  CD8ABB    CALL BB8A     ;afficher curseur
872   95D0  C9        RET           ;

      ** SOUS-PROGRAMME 33 **

873   95D1  0601      LD B,01       ;B=1 pour scrolling vers le haut
874   95D3  210100    LD HL,0001    ;preparer registres
875   95D6  11164F    LD DE,4F16    ;
876   95D9  3E00      LD A,00       ;
877   95DB  CD50BC    CALL BC50     ;scrolling
878   95DE  C9        RET           ;

      ** SOUS-PROGRAMME 34 **

879   95DF  0600      LD B,00       ;B=0 pour scrolling vers le bas
880   95E1  210100    LD HL,0001    ;preparer registres
881   95E4  11164F    LD DE,4F16    ;
882   95E7  3E00      LD A,00       ;
883   95E9  CD50BC    CALL BC50     ;scrolling
884   95EC  C9        RET           ;
```

```
      ** SOUS-PROGRAMME 35 **

885   95ED  AF        XOR A         ;enlever carry et A=0
886   95EE  E5        PUSH HL       ;preserver HL
887   95EF  ED52      SBC HL,DE     ;
888   95F1  E1        POP HL        ;recuperer HL
889   95F2  C8        RET Z         ;retour si HL-DE=0
890   95F3  3803      JRC 95F8      ;report, alors DE > HL
891   95F5  3E01      LD A,01       ;DE < HL, indicateur a 1...
892   95F7  C9        RET           ;... et retour
893   95F8  AF        XOR A         ;DE > HL, enlever carry
894   95F9  3EFF      LD A,FF       ;indicateur a 255...
895   95FB  C9        RET           ;... et retour

      ** SOUS-PROGRAMME 36 **

896   95FC  FE00      CP 00         ;afficher colonne seule ?
897   95FE  2814      JRZ 9614      ;oui
898   9600  F5        PUSH AF       ;non, preserver AF
899   9601  211904    LD HL,0419    ;curseur systeme en (4,25)
900   9604  CD75BB    CALL BB75     ;
901   9607  3A853A    LD A,(3A85)   ;ligne theorique dans A
902   960A  6F        LD L,A        ;puis dans HL
903   960B  2600      LD H,00       ;
904   960D  CD2496    CALL 9624     ;affichage ligne theorique
905   9610  F1        POP AF        ;recuperer AF
```

```
906   9611  FE01      CP 01          ;afficher aussi colonne ?
907   9613  C0        RET NZ         ;non, alors fini
908   9614  21190A    LD HL,0A19     ;oui, alors curseur systeme en (10,25)
909   9617  CD75BB    CALL BB75      ;
910   961A  3A863A    LD A,(3A86)    ;colonne dans A
911   961D  6F        LD L,A         ;puis dans HL
912   961E  2600      LD H.00        ;
913   9620  CD2496    CALL 9624      ;affichage
914   9623  C9        RET            ;retour de sous-programme
915   9624  DF        RST 24         ;* AFFICHAGE
916   9625  2D96                     ;adresse bloc de parametres
917   9627  3E20      LD A,20        ;affichage termine...
918   9629  CD5ABB    CALL BB5A      ;...afficher un blanc supplementaire
919   962C  C9        RET            ;retour de RESTART
920   962D  79EE                     ;adresse ROM
921   962F  FC                       ;octet de selection de la ROM

      ** EFFACER JUSQU'AU CURSEUR **

922   963A  CD3495    CALL 9534      ;curseur memoire = fin de texte ?
923   963D  CAAC96    JP Z,96AC      ;oui, alors effacer tout
924   9640  D5        PUSH DE        ;DE contient cursr. memoire, preserver
925   9641  2A8A3A    LD HL,(3A8A)   ;fin de texte dans HL
926   9644  AF        XOR A          ;enlever carry
927   9645  ED52      SBC HL,DE      ;nbre d'octets a transferer (= ot) ds HL
928   9647  23        INC HL         ;
```

```
929   9648  E5        PUSH HL       ;preserver
930   9649  11973A    LD DE,3A97    ;nouvelle fin de texte = 14999 + ot
931   964C  19        ADD HL,DE     ;
932   964D  228A3A    LD (3A8A),HL  ;ranger dans placard
933   9650  C1        POP BC        ;ot dans BC
934   9651  E3        EX (SP),HL    ;cursr. mem. ds HL et nlle f.d.t. sur pile
935   9652  11983A    LD DE,3A98    ;preparer DE pour transfert
936   9655  EDB0      LDIR          ;transfert
937   9657  CD8096    CALL 9680     ;modifier variables
938   965A  E1        POP HL        ;fin de texte dans HL
939   965B  CD7096    CALL 9670     ;nettoyer zone (HL a 39400)
940   965E  21983A    LD HL,3A98    ;HL pointe dur debut RAM (15000)
941   9661  117841    LD DE,4178    ;afficher ecran (1760 caracteres)
942   9664  CD8A94    CALL 948A     ;
943   9667  3E01      LD A,01       ;afficher nlle ligne et nlle colonne
944   9669  CDFC95    CALL 95FC     ;
945   966C  CDCA95    CALL 95CA     ;afficher curseur
946   966F  C9        RET           ;
```

** SOUS-PROGRAMME 37 **

```
947   9670  E5        PUSH HL       ;preserver HL
948   9671  36FF      LD (HL),FF    ;code espace inutile dans (HL)
949   9673  23        INC HL        ;HL sur adresse suivante
950   9674  3E8A      LD A,8A       ;H = octet fort de 35400 ?
951   9676  BC        CP H          ;
```

```
952   9677  20F8      JRNZ 9671     ;non, continuer boucle
953   9679  3E48      LD A,48       ;oui, et L = octet faible de 35400 ?
954   967B  BD        CP L          ;
955   967C  20F3      JRNZ 9671     ;non, continuer boucle
956   967E  E1        POP HL        ;oui, recuperer HL
957   967F  C9        RET           ;et retour

      ** SOUS-PROGRAMME 38 **

958   9680  E5        PUSH HL       ;preserver HL
959   9681  21843A    LD HL,3A84    ;HL pointe sur placard ligne ecran
960   9684  3602      LD (HL),02    ;ligne ecran = 2
961   9686  23        INC HL        ;HL pointe sur placard ligne theorique
962   9687  3601      LD (HL),01    ;ligne theorique = 1
963   9689  23        INC HL        ;HL pointe sur placard colonne
964   968A  3601      LD (HL),01    ;colonne = 1
965   968C  21983A    LD HL,3A98    ;15000 dans HL
966   968F  22883A    LD (3A88),HL  ;curseur memoire = 15000
967   9692  E1        POP HL        ;recuperer HL
968   9693  C9        RET           ;

      ** EFFACER A PARTIR DU CURSEUR **

969   9694  CD3495    CALL 9534     ;curseur memoire = fin de texte ?
970   9697  C8        RET Z         ;oui, fini
```

```
971   9698  EB        EX DE,HL      ;HL=f.d.t.-cursr. mem. et DE=cusr. mem.
972   9699  228A3A    LD (3A8A),HL  ;nlle fin de texte=curseur memoire
973   969C  CD7096    CALL 9670     ;nettoyer zone (HL a 39400)
974   969F  EB        EX DE,HL      ;DE=f.d.t.-cursr. mem. et HL=cusr. mem.
975   96A0  01E006    LD BC,06E0    ;
976   96A3  19        ADD HL,DE     ;HL=ancienne fin de texte
977   96A4  EB        EX DE,HL      ;DE=ancienne f.d.t et HL=nlle f.d.t
978   96A5  CD8A94    CALL 948A     ;afficher zone (HL) a (DE)
979   96A8  CDCA95    CALL 95CA     ;afficher curseur
980   96AB  C9        RET           ;
```

** EFFACER TOUT **

```
981   96AC  21983A    LD HL,3A98    ;15000 dans HL
982   96AF  228A3A    LD (3A8A),HL  ;fin de texte = 15000
983   96B2  CD8096    CALL 9680     ;modifier variables
984   96B5  3E01      LD A,01       ;afficher nlle ligne et nlle colonne
985   96B7  CDFC95    CALL 95FC     ;
986   96BA  C3948E    JP 8E94       ;* VERS INITIALISATION
```

** RANGER CURSEUR MEMOIRE AVANT EFFACAGE **

```
987   96BD  3A843A    LD A,(3A84)   ;ligne ecran dans A
988   96C0  47        LD B,A        ;puis dans B
989   96C1  3A863A    LD A,(3A86)   ;colonne dans A
```

```
990   96C4  2A883A    LD HL,(3A88) ;curseur memoire dans HL
991   96C7  3D        DEC A        ;decrementer colonne
992   96C8  2803      JRZ 96CD     ;saut si 0
993   96CA  2B        DEC HL       ;sinon, decrementer curseur memoire...
994   96CB  18FA      JR 96C7      ;... et continuer boucle
995   96CD  3E02      LD A,02      ;ligne ecran = 2 ?
996   96CF  B8        CP B         ;
997   96D0  2004      JRNZ 96D6    ;non, continuer
998   96D2  22823A    LD (3A82),HL ;crsr. mem. corresp. a debut page/ranger
999   96D5  C9        RET          ;
1000  96D6  05        DEC B        ;ligne ecran -1...
1001  96D7  3E51      LD A,51      ;donc colonne passe a 81
1002  96D9  18EC      JR 96C7      ;boucler

      ** REAFFICHAGE DE L'ECRAN **

1003  96DB  3A843A    LD A,(3A84)  ;ligne ecran dans A
1004  96DE  F5        PUSH AF      ;ranger
1005  96DF  3A863A    LD A,(3A86)  ;colonne dans A
1006  96E2  F5        PUSH AF      ;ranger
1007  96E3  3E01      LD A,01      ;colonne = 1
1008  96E5  32863A    LD (3A86),A  ;
1009  96E8  3E02      LD A,02      ;ligne ecran = 2
1010  96EA  32843A    LD (3A84),A  ;
1011  96ED  2A823A    LD HL,(3A82) ;recuperer curseur memoire
1012  96F0  E5        PUSH HL      ;ranger
1013  96F1  11E006    LD DE,06E0   ;HL=curseur memoire+DE (+1760)
```

```
1014  96F4  19        ADD HL,DE     ;
1015  96F5  EB        EX DE,HL      ;DE=curseur memoire+1760
1016  96F6  E1        POP HL        ;HL=curseur memoire
1017  96F7  CD8A94    CALL 948A     ;afficher zone (HL) a (DE)
1018  96FA  F1        POP AF        ;recuperer colonne
1019  96FB  32863A    LD (3A86),A   ;ranger dans placard
1020  96FE  F1        POP AF        ;recuperer ligne ecran
1021  96FF  32843A    LD (3A84),A   ;ranger dans placard
1022  9702  CDCA95    CALL 95CA     ;afficher curseur


      ** NETTOYAGE COMPLET DE LA RAM **

1023  9706  21983A    LD HL,3A98    ;15000 dans HL
1024  9709  CD7096    CALL 9670     ;nettoyer zone (HL a 39400)
1025  970C  C9        RET           ;


      ** AFFICHAGE ECRAN APRES CHARGEMENT D'UN TEXTE **

1026  970D  CD8096    CALL 9680     ;modifier variables
1027  9710  C35E96    JP 965E       ;* SAUT EN LIGNE 940

                  ****************************************
                  ****************************************
```

```
      ** NETTOYAGE DU BUFFER CLAVIER **

1028  975E  CD1BBB    CALL BB1B      ;touche disponible ?
1029  9761  38FB      JRC 975E       ;oui, alors interroger a nouveau
1030  9763  C9        RET            ;non, fini


      ** AFFICHER A PARTIR D'UNE PAGE SPECIFIEE **

1031  9764  CD8497    CALL 9784      ;transformer 'SORTIE ECRAN' en sp
1032  9767  3A4197    LD A,(9741)    ;numero 1ere page dans A
1033  976A  47        LD B,A         ;puis dans B
1034  976B  3A4297    LD A,(9742)    ;page actuelle dans A
1035  976E  B8        CP B           ;page actuelle=page specifiee ?
1036  976F  2006      JRNZ 9777      ;non
1037  9771  CDA897    CALL 97A8      ;oui, restaurer 'SORTIE ECRAN'
1038  9774  C3D997    JP 97D9        ;* VERS SORTIE ECRAN
1039  9777  C5        PUSH BC        ;preserver numero 1ere page
1040  9778  CDE197    CALL 97E1      ;appeler 'SORTIE ECRAN' comme un sp
1041  977B  C1        POP BC         ;recuperer numero 1ere page
1042  977C  FE01      CP 01          ;affichage termine ?
1043  977E  C8        RET Z          ;oui, fini
1044  977F  FE04      CP 04          ;fin de texte atteinte ?
1045  9781  C8        RET Z          ;oui, fini
1046  9782  18E7      JR 976B        ;boucler
```

```
        ** SOUS-PROGRAMME 39 **

1047   9784   215998    LD HL,9859    ;&C9 en 39001
1048   9787   36C9      LD. (HL),C9   ;
1049   9789   21D997    LD HL,97D9    ;&C9 en 38873
1050   978C   36C9      LD (HL),C9    ;
1051   978E   218498    LD HL,9884    ;Ø en 39044
1052   9791   AF        XOR A         ;
1053   9792   77        LD (HL),A     ;
1054   9793   23        INC HL        ;Ø en 39045
1055   9794   77        LD (HL),A     ;
1056   9795   23        INC HL        ;Ø en 39046
1057   9796   77        LD (HL),A     ;
1058   9797   21339A    LD HL,9A33    ;Ø en 39475
1059   979A   77        LD (HL),A     ;
1060   979B   23        INC HL        ;Ø en 39476
1061   979C   77        LD (HL),A     ;
1062   979D   23        INC HL        ;Ø en 39477
1063   979E   77        LD (HL),A     ;
1064   979F   21609A    LD HL,9A60    ;Ø en 39520
1065   97A2   77        LD (HL),A     ;
1066   97A3   23        INC HL        ;Ø en 39521
1067   97A4   77        LD (HL),A     ;
1068   97A5   23        INC HL        ;Ø en 39522
1069   97A6   77        LD (HL),A     ;
1070   97A7   C9        RET           ;
```

```
      ** SOUS-PROGRAMME 40 **

1071  97A8  215998    LD HL,9859    ;* RESTAURER SOUS-PROGRAMME 44
1072  97AB  36F5      LD (HL),F5    ;
1073  97AD  21D997    LD HL,97D9    ;* RESTAURER 'SORTIE ECRAN'
1074  97B0  36CD      LD (HL),CD    ;
1075  97B2  218498    LD HL,9884    ;* RESTAURER SOUS-PROGRAMME 45
1076  97B5  36CD      LD (HL),CD    ;
1077  97B7  23        INC HL        ;
1078  97B8  365A      LD (HL),5A    ;
1079  97BA  23        INC HL        ;
1080  97BB  36BB      LD (HL),BB    ;
1081  97BD  21339A    LD HL,9A33    ;* RESTAURER SOUS-PROGRAMME 55
1082  97C0  36CD      LD (HL),CD    ;
1083  97C2  23        INC HL        ;
1084  97C3  365A      LD (HL),5A    ;
1085  97C5  23        INC HL        ;
1086  97C6  36BB      LD (HL),BB    ;
1087  97C8  21609A    LD HL,9A60    ;
1088  97CB  36CD      LD (HL),CD    ;
1089  97CD  23        INC HL        ;
1090  97CE  365A      LD (HL),5A    ;
1091  97D0  23        INC HL        ;
1092  97D1  36BB      LD (HL),BB    ;
1093  97D3  C9        RET           ;
```

```
      ** SORTIE ECRAN **

1094  97D9  CD1BBB   CALL BB1B     ;interrogation clavier
1095  97DC  30FB     JRNC 97D9     ;aucune touche pressee, reinterroger
1096  97DE  FE8B     CP 8B         ;petite touche ENTER ?
1097  97E0  C8       RET Z         ;oui, fini
1098  97E1  3A8E3A   LD A,(3A8E)   ;parametre 'marge haute' dans A
1099  97E4  FE00     CP 00         ;egal a 0 ?
1100  97E6  2809     JRZ 97F1      ;oui
1101  97E8  47       LD B,A        ;non, alors marge haute dans B
1102  97E9  CD4A98   CALL 984A     ;effectuer B sauts de ligne
1103  97EC  21E697   LD HL,97E6   ;mettre hors service partie 'marge haute
1104  97EF  3618     LD (HL),18    ;
1105  97F1  CD3D98   CALL 983D     ;calculer nbre caracteres par ligne
1106  97F4  CD2F98   CALL 982F     ;curseur systeme en fonction marge gauche
1107  97F7  2A4597   LD HL,(9745) ;adresse ligne actuelle dans HL
1108  97FA  E5       PUSH HL       ;preserver
1109  97FB  CDD298   CALL 98D2     ;calculer nbre de mots et nbre de lettres
1110  97FE  224797   LD (9747),HL ;adresse prochaine ligne en (38727)
1111  9801  E1       POP HL        ;recuperer adresse ligne actuelle
1112  9802  FE06     CP 06         ;mot trop long pour les marges ?
1113  9804  2006     JRNZ 980C     ;non
1114  9806  3E01     LD A,01       ;oui, mettre semaphore
1115  9808  324D97   LD (974D),A   ;
1116  980B  C9       RET           ;fini
1117  980C  FE01     CP 01         ;plus rien a faire ?
1118  980E  C8       RET Z         ;non, alors fini
```

```
1119  980F  FE02      CP 02          ;simple saut de ligne ?
1120  9811  2806      JRZ 9819       ;oui
1121  9813  CDDF99    CALL 99DF      ;non, sortir ligne
1122  9816  FE04      CP 04          ;fin de texte atteinte ?
1123  9818  C8        RET Z          ;oui, alors fini
1124  9819  0601      LD B,01        ;simple saut de ligne: 1 dans B
1125  981B  CD4A98    CALL 984A      ;sauter B lignes
1126  981E  CD9C98    CALL 989C      ;si interligne=2, sauter encore 1 ligne
1127  9821  CDAD98    CALL 98AD      ;sauter eventuelles lignes c.d.c.
1128  9824  2A4797    LD HL,(9747)   ;adresse prochaine ligne dans HL
1129  9827  224597    LD (9745),HL   ;prochaine ligne devient ligne actuelle
1130  982A  18AD      JR 97D9        ;reprendre

      ** SOUS-PROGRAMME 41 **

1131  982F  E5        PUSH HL        ;preserver HL
1132  9830  3A8C3A    LD A,(3A8C)    ;marge gauche dans A
1133  9833  67        LD H,A         ;puis dans H
1134  9834  3A4997    LD A,(9749)    ;ligne ecran dans A
1135  9837  6F        LD L,A         ;puis dans L
1136  9838  CD75BB    CALL BB75      ;positionner curseur systeme en (H,L)
1137  983B  E1        POP HL         ;recuperer HL
1138  983C  C9        RET            ;
```

```
      ** SOUS-PROGRAMME 42 **

1139  983D  3A8C3A    LD A,(3A8C)   ;marge gauche dans A
1140  9840  47        LD B,A        ;puis dans B
1141  9841  3A8D3A    LD A,(3A8D)   ;marge droite dans A
1142  9844  90        SUB B         ;marge droite-marge gauche dans A
1143  9845  3C        INC A         ;nombre de caracteres par ligne dans A
1144  9846  324A97    LD (974A),A   ;ranger
1145  9849  C9        RET           ;

      ** SOUS-PROGRAMME 43 **

1146  984A  E5        PUSH HL       ;preserver registres
1147  984B  F5        PUSH AF       ;
1148  984C  C5        PUSH BC       ;
1149  984D  CD5998    CALL 9859     ;passer a la ligne suivante
1150  9850  CD7298    CALL 9872     ;decrementer compteur de lignes
1151  9853  C1        POP BC        ;recuperer BC
1152  9854  10F6      DJNZ 984C     ;si B <> 0 ,boucler (B=nbre lign. a sauter)
1153  9856  F1        POP AF        ;recuperer registres
1154  9857  E1        POP HL        ;
1155  9858  C9        RET           ;
```

```
      ** SOUS-PROGRAMME 44 **

1156  9859  F5        PUSH AF        ;preserver AF
1157  985A  3A4997    LD A,(9749)    ;numero ligne ecran actuelle dans A
1158  985D  3C        INC A          ;ligne ecran + 1
1159  985E  FE18      CP 18          ;ligne ecran = 24 ?
1160  9860  200B      JRNZ 986D      ;non
1161  9862  E5        PUSH HL        ;oui, alors preserver registres...
1162  9863  C5        PUSH BC        ;
1163  9864  D5        PUSH DE        ;
1164  9865  CDD195    CALL 95D1      ;...scrolling ecran vers le haut...
1165  9868  D1        POP DE         ;...recuperer registres...
1166  9869  C1        POP BC         ;
1167  986A  E1        POP HL         ;
1168  986B  3E17      LD A,17        ;...et ligne ecran=23
1169  986D  324997    LD (9749),A    ;ranger nouvelle ligne ecran
1170  9870  F1        POP AF         ;recuperer A
1171  9871  C9        RET            ;

      ** SOUS-PROGRAMME 45 **

1172  9872  214497    LD HL,9744     ;HL pointe sur compteur de lignes
1173  9875  35        DEC (HL)       ;compteur de ligne -1
1174  9876  C0        RET NZ         ;si compteur <> 0, alors fini
1175  9877  3A4997    LD A,(9749)    ;ligne ecran actuelle dans A
1176  987A  6F        LD L,A         ;puis dans L
```

```
1177  987B  2602      LD H,02        ;
1178  987D  CD75BB    CALL BB75      ;positionner curseur systeme en (H,L)
1179  9880  064E      LD B,4E        ;* ECRIRE LIGNE POINTILLES(78 fois code 45)
1180  9882  3E2D      LD A,2D        ;
1181  9884  CD5ABB    CALL BB5A      ;
1182  9887  10FB      DJNZ 9884      ;
1183  9889  CD5998    CALL 9859      ;passer a la ligne suivante
1184  988C  3A903A    LD A.(3A90)    ;restaurer compteur de lignes
1185  988F  324497    LD (9744),A    ;
1186  9892  21E697    LD HL,97E6     ;deverouiller 'marge haute' de SORTIE ECRAN
1187  9895  3628      LD (HL),28     ;
1188  9897  214297    LD HL,9742     ;numero de page + 1
1189  989A  34        INC (HL)       ;
1190  989B  C9        RET            ;
```

** SOUS-PROGRAMME 46 **

```
1191  989C  F5        PUSH AF        ;preserver AF
1192  989D  3A8F3A    LD A,(3A8F)    ;parametre 'interligne' dans A
1193  98A0  FE02      CP 02          ;= 2 ?
1194  98A2  2802      JRZ 98A6       ;oui
1195  98A4  F1        POP AF         ;non, alors recuperer A et retour
1196  98A5  C9        RET            ;
1197  98A6  0601      LD B,01        ;sauter une ligne
1198  98A8  CD4A98    CALL 984A      ;
1199  98AB  F1        POP AF         ;recuper A et retour
1200  98AC  C9        RET            ;
```

```
      ** SOUS-PROGRAMME 47 **

1201  98AD  3A4397    LD A,(9743)   ;nombre de lignes a sauter dans A
1202  98B0  FE00      CP 00         ;= 0 ?
1203  98B2  C8        RET Z         ;oui, retour immediat
1204  98B3  47        LD B,A        ;nombre de lignes dans B
1205  98B4  3A4097    LD A,(9740)   ;numero du peripherique dans A
1206  98B7  FE01      CP 01         ;= 1 ?
1207  98B9  2805      JRZ 98C0      ;oui, alors vers partie imprimante
1208  98BB  CD4A98    CALL 984A     ;sauter B lignes ecran
1209  98BE  1803      JR 98C3       ;vers fin commune
1210  98C0  CD3F9B    CALL 9B3F     ;sauter B lignes imprimante
1211  98C3  AF        XOR A         ;remettre placard 38723 a 0
1212  98C4  324397    LD (9743),A   ;
1213  98C7  C9        RET           ;

      ** SOUS-PROGRAMME 48 **

1214  98D2  010000    LD BC,0000    ;B et C a 0
1215  98D5  3A4A97    LD A,(974A)   ;nombre de caracteres par ligne dans A
1216  98D8  57        LD D,A        ;puis dans D (compteur general)
1217  98D9  CDD099    CALL 99D0     ;HL >= a fin de texte ?
1218  98DC  FEFF      CP FF         ;
1219  98DE  280A      JRZ 98EA      ;non
1220  98E0  AF        XOR A         ;oui, et nombre de mots (B)= 0 ?
1221  98E1  B8        CP B          ;
```

```
1222  98E2  2003      JRNZ 98E7     ;non
1223  98E4  3E01      LD A,01       ;oui, signal 1 dans A et retour
1224  98E6  C9        RET           ;
1225  98E7  3E04      LD A,04       ;signal 4 dans A et retour
1226  98E9  C9        RET           ;
1227  98EA  3E20      LD A,20       ;* IGNORER ESPAC. INUT. ET ESPAC. BLCS
1228  98EC  BE        CP (HL)       ;
1229  98ED  2003      JRNZ 98F2     ;
1230  98EF  23        INC HL        ;
1231  98F0  18E7      JR 98D9       ;
1232  98F2  3EFF      LD A,FF       ;
1233  98F4  BE        CP (HL)       ;
1234  98F5  2003      JRNZ 98FA     ;
1235  98F7  23        INC HL        ;
1236  98F8  18DF      JR 98D9       ;
1237  98FA  3EF1      LD A,F1       ;* CODE DE CONTROLE ?
1238  98FC  BE        CP (HL)       ;
1239  98FD  2005      JRNZ 9904     ;non
1240  98FF  CD3999    CALL 9939     ;oui, executer...
1241  9902  18D5      JR 98D9       ;...et continuer
1242  9904  3EF2      LD A,F2       ;* FIN DE PARAGRAPHE ?
1243  9906  BE        CP (HL)       ;
1244  9907  200B      JRNZ 9914     ;non
1245  9909  23        INC HL        ;oui, pointer HL sur adresse suivante
1246  990A  AF        XOR A         ;nombre de mots (B)= 0 ?
1247  990B  B8        CP B          ;
1248  990C  2003      JRNZ 9911     ;non
```

```
1249  990E  3E02      LD A,02       ;oui, alors signal 2 dans A et retour
1250  9910  C9        RET           ;
1251  9911  3E03      LD A,03       ;non, alors signal 3 dans A et retour
1252  9913  C9        RET           ;
1253  9914  E5        PUSH HL       ;* PAS ESPACE BLANC, NI ESPACE  INUTILE...
1254  9915  C5        PUSH BC       ;NI CODE DE CTRL, NI FIN PARAGR. ...
1255  9916  D5        PUSH DE       ;ALORS CARACT. / PRE SERVER REGISTRES
1256  9917  CD9299    CALL 9992     ;mot correspondant entre-t-il ds ligne?
1257  991A  FE01      CP 01         ;mot trop long ?
1258  991C  2005      JRNZ 9923     ;non
1259  991E  D1        POP DE        ;oui, recuperer registres...
1260  991F  C1        POP BC        ;
1261  9920  E1        POP HL        ;
1262  9921  180C      JR 992F       ;... et saut
1263  9923  AF        XOR A         ;compteur general (D)=0 ?
1264  9924  BA        CP D          ;
1265  9925  2805      JRZ 992C      ;oui
1266  9927  F1        POP AF        ;non, nettoyer pile...
1267  9928  F1        POP AF        ;
1268  9929  F1        POP AF        ;
1269  992A  18AD      JR 98D9       ;... et continuer
1270  992C  F1        POP AF        ;oui, nettoyer pile
1271  992D  F1        POP AF        ;
1272  992E  F1        POP AF        ;
1273  992F  AF        XOR A         ;nombre de mots (B)=0 ?
1274  9930  B8        CP B          ;
1275  9931  2803      JRZ 9936      ;oui, alors signal 6 dans A et retour
```

```
1276  9933  3E05      LD A,05       ;non, alors signal 5 dans A et retour
1277  9935  C9        RET           ;
1278  9936  3E06      LD A,06       ;
1279  9938  C9        RET           ;

      ** SOUS-PROGRAMME 49 **

1280  9939  E5        PUSH HL       ;preserver registres
1281  993A  D5        PUSH DE       ;
1282  993B  C5        PUSH BC       ;
1283  993C  F5        PUSH AF       ;
1284  993D  23        INC HL        ;HL pointe sur 1er code ASCII
1285  993E  3E30      LD A,30       ;= 0 ? (code ASCII de 0=&30)
1286  9940  BE        CP (HL)       ;
1287  9941  2045      JRNZ 9988     ;non, alors ignorer
1288  9943  23        INC HL        ;HL pointe sur 3eme code ASCII
1289  9944  23        INC HL        ;
1290  9945  7E        LD A,(HL)     ;code ASCII dans A
1291  9946  D630      SUB 30        ;valeur correspondante ds A
1292  9948  0609      LD B,09       ;* MULTIPLIER CETTE VALEUR PAR 10
1293  994A  4F        LD C,A        ;
1294  994B  81        ADD A,C       ;
1295  994C  10FD      DJNZ 994B     ;
1296  994E  47        LD B,A        ;resultat dans B
1297  994F  23        INC HL        ;HL pointe sur 4eme code ASCII
1298  9950  7E        LD A,(HL)     ;code ASCII dans A
```

```
1299  9951  D630      SUB 30         ;valeur correspondante dans A
1300  9953  80        ADD A,B        ;unites plus dizaines dans A
1301  9954  4F        LD C,A         ;preserver provisoirement dans C
1302  9955  2B        DEC HL         ;HL pointe sur 2eme code ASCII
1303  9956  2B        DEC HL         ;
1304  9957  7E        LD A,(HL)      ;code ASCII dans A
1305  9958  FE31      CP 31          ;code de saut de ligne ?
1306  995A  2825      JRZ 9981       ;oui
1307  995C  FE32      CP 32          ;non, alors code de marge gauche ?
1308  995E  2811      JRZ 9971       ;oui
1309  9960  3A8C3A    LD A,(3A8C)    ;* NON, DONC MARGE DROITE / MARGE DROITE
1310  9963  47        LD B,A         ;MOINS MARGE GAUCHE + 1 >=15?
1311  9964  04        INC B          ;
1312  9965  79        LD A,C         ;
1313  9966  90        SUB B          ;
1314  9967  FE0F      CP 0F          ;
1315  9969  381D      JRC 9988       ;non, vers fin commune
1316  996B  79        LD A,C         ;oui, alors nouvelle marge droite dans A...
1317  996C  328D3A    LD (3A8D),A    ;...ranger...
1318  996F  1817      JR 9988        ;...et vers fin commune
1319  9971  41        LD B,C         ;* MARGE DROITE-MARGE GAUCHE+1 >= 15 ?
1320  9972  3A8D3A    LD A,(3A8D)    ;
1321  9975  3C        INC A          ;
1322  9976  90        SUB B          ;
1323  9977  FE0F      CP 0F          ;
1324  9979  380D      JRC 9988       ;non, vers fin commune
1325  997B  79        LD A,C         ;oui, alors nouvelle marge gauche dans A...
```

```
1326  997C  328C3A    LD (3A8C),A   ;...ranger...
1327  997F  1807      JR 9988       ;...et vers fin commune
1328  9981  3A4397    LD A,(9743)   ;compteur saut de ligne dans A
1329  9984  81        ADD A,C       ;+ nouveaux sauts de ligne
1330  9985  324397    LD (9743),A   ;ranger
1331  9988  F1        POP AF        ;recuperer registres
1332  9989  C1        POP BC        ;
1333  998A  D1        POP DE        ;
1334  998B  E1        POP HL        ;
1335  998C  23        INC HL        ;pointer HL sur octet apres code de ctrl.
1336  998D  23        INC HL        ;
1337  998E  23        INC HL        ;
1338  998F  23        INC HL        ;
1339  9990  23        INC HL        ;
1340  9991  C9        RET           ;fini
```

** SOUS-PROGRAMME 50 **

```
1341  9992  04        INC B         ;compteur de mots + 1
1342  9993  15        DEC D         ;compteur general - 1
1343  9994  1814      JR 99AA       ;
1344  9996  23        INC HL        ;HL pointe sur adresse suivante
1345  9997  0C        INC C         ;compteur de lettres + 1
1346  9998  15        DEC D         ;compteur general - 1
1347  9999  7E        LD A,(HL)     ;
1348  999A  FEFF      CP FF         ;HL pointe sur espace inutile ?
```

```
1349  999C  2818      JRZ 99B6       ;oui alors mot termine, vers fin commune
1350  999E  FE20      CP 20          ;HL pointe sur espace blanc ?
1351  99A0  2814      JRZ 99B6       ;oui alors mot termine, vers fin commune
1352  99A2  FEF1      CP F1          ;HL pointe sur code de controle ?
1353  99A4  2810      JRZ 99B6       ;oui alors mot termine, vers fin commune
1354  99A6  FEF2      CP F2          ;HL pointe sur fin de paragraphe ?
1355  99A8  2807      JRZ 99B1       ;oui alors mot termine, vers fin speciale
1356  99AA  AF        XOR A          ;mot non termine / compteur general=0 ?
1357  99AB  BA        CP D           ;
1358  99AC  20E8      JRNZ 9996      ;non, continuer
1359  99AE  3E01      LD A,01        ;oui, signal mot trop long dans A...
1360  99B0  C9        RET            ;...et retour
1361  99B1  AF        XOR A          ;mot termine / compteur gal=0 ?
1362  99B2  BA        CP D           ;
1363  99B3  2001      JRNZ 99B6      ;non
1364  99B5  23        INC HL         ;oui, alors eviter fin de paragraphe
1365  99B6  AF        XOR A          ;signal mot pas trop long dans A...
1366  99B7  C9        RET            ;et retour
```

** SOUS-PROGRAMME 51 **

```
1367  99B8  E5        PUSH HL        ;preserver HL
1368  99B9  3A4A97    LD A,(974A)    ;nombre de caracteres par ligne dans A
1369  99BC  91        SUB C          ;nombre total de blancs dans A
1370  99BD  6F        LD L,A         ;puis dans HL
1371  99BE  2600      LD H,00        ;
```

```
1372  99C0  48        LD C,B         ;nombre de mots-1 dans BC
1373  99C1  0600      LD B,00        ;
1374  99C3  0D        DEC C          ;
1375  99C4  CD7C91    CALL 917C      ;division HL/BC
1376  99C7  324B97    LD (974B),A    ;ranger nombre de blancs entre chaque mot
1377  99CA  7D        LD A,L         ;nombre de blancs en plus dans A
1378  99CB  324C97    LD (974C),A    ;ranger
1379  99CE  E1        POP HL         ;recuperer HL
1380  99CF  C9        RET            ;
```

** SOUS-PROGRAMME 52 **

```
1381  99D0  D5        PUSH DE        ;preserver DE
1382  99D1  ED5B8A3A  LD DE,(3A8A)   ;fin de texte dans HL
1383  99D5  CDED95    CALL 95ED      ;comparer HL et DE
1384  99D8  D1        POP DE         ;recuperer DE
1385  99D9  C9        RET            ;
```

** SOUS-PROGRAMME 53 **

```
1386  99DF  F5        PUSH AF        ;preserver AF
1387  99E0  78        LD A,B         ;nombre de mots dans A
1388  99E1  FE01      CP 01          ;1 seul mot ?
1389  99E3  2003      JRNZ 99E8      ;non
1390  99E5  F1        POP AF         ;oui
```

```
1391  99E6  180E      JR 99F6        ;
1392  99E8  F1        POP AF         ;recuperer AF
1393  99E9  FE05      CP 05          ;ligne normale ?
1394  99EB  2009      JRNZ 99F6      ;non
1395  99ED  F5        PUSH AF        ;oui, preserver AF
1396  99EE  CDB899    CALL 99B8      ;calculer nbre de blancs entre chaque mot
1397  99F1  CD0B9A    CALL 9A0B      ;sortir ligne
1398  99F4  F1        POP AF         ;recuperer A
1399  99F5  C9        RET            ;fini
1400  99F6  F5        PUSH AF        ;1 seul mot, ou ligne se termine par...
1401  99F7  79        LD A,C         ;f.d.t., ou fin paragraphe; donc...
1402  99F8  80        ADD A,B        ;nbre de caract. par ligne=nbre lettres...
1403  99F9  3D        DEC A          ;+ nombre mots - 1 , nombre de blancs...
1404  99FA  324A97    LD (974A),A    ;entre chaque mot=1, et blancs...
1405  99FD  3E01      LD A,01        ;supplementaires=0
1406  99FF  324B97    LD (974B),A    ;
1407  9A02  AF        XOR A          ;
1408  9A03  324C97    LD (974C),A    ;
1409  9A06  CD0B9A    CALL 9A0B      ;sortir ligne ecran
1410  9A09  F1        POP AF         ;recuperer AF
1411  9A0A  C9        RET            ;
```

** SOUS-PROGRAMME 54 **

```
1412  9A0B  3A4A97    LD A,(974A)    ;nombre de caracteres par ligne dans A
1413  9A0E  47        LD B,A         ;puis dans B
```

```
1414  9A0F  7E        LD A,(HL)     ;* IGNORER ESPACES INUTILES
1415  9A10  FEFF      CP FF         ;
1416  9A12  2003      JRNZ 9A17     ;
1417  9A14  23        INC HL        ;
1418  9A15  18F8      JR 9A0F       ;
1419  9A17  FE20      CP 20         ;* IGNORER ESPACES BLANCS
1420  9A19  2003      JRNZ 9A1E     ;
1421  9A1B  23        INC HL        ;
1422  9A1C  18F1      JR 9A0F       ;
1423  9A1E  FEF1      CP F1         ;ni l'un ni l'autre, alors c.d.c ?
1424  9A20  2005      JRNZ 9A27     ;non
1425  9A22  CD689A    CALL 9A68     ;oui, alors executer si canal imprimante...
1426  9A25  18E8      JR 9A0F       ;et continuer
1427  9A27  CD2D9A    CALL 9A2D     ;caract. trouve, afficher mot correspondant
1428  9A2A  C8        RET Z         ;si compteur general=0, alors fini...
1429  9A2B  18E2      JR 9A0F       ;sinon continuer
```

** SOUS-PROGRAMME 55 **

```
1430  9A2D  FEF3      CP F3         ;espace force ?
1431  9A2F  2002      JRNZ 9A33     ;non
1432  9A31  3E20      LD A,20       ;oui, alors code espace dans A
1433  9A33  CD5ABB    CALL BB5A     ;afficher caractere
1434  9A36  05        DEC B         ;decrementer compteur general
1435  9A37  C8        RET Z         ;retour si 0
1436  9A38  23        INC HL        ;pointer HL sur caractere suivant
```

```
1437  9A39  7E       LD A,(HL)     ;caractere dans A
1438  9A3A  FEF1     CP F1         ;code de controle ?
1439  9A3C  2808     JRZ 9A46      ;oui, alors mot termine
1440  9A3E  FEFF     CP FF         ;espace inutile ?
1441  9A40  2804     JRZ 9A46      ;oui, alors mot termine
1442  9A42  FE20     CP 20         ;espace blanc ?
1443  9A44  20E7     JRNZ 9A2D     ;non, alors mot pas termine, continuer
1444  9A46  3A4B97   LD A,(974B)   ;nombre de blancs entre chaque mot dans A
1445  9A49  57       LD D,A        ;puis dans D
1446  9A4A  3E20     LD A,20       ;code espace dans A
1447  9A4C  CD5ABB   CALL BB5A     ;afficher un blanc
1448  9A4F  05       DEC B         ;decrementer compteur general
1449  9A50  15       DEC D         ;decrementer compteur de blancs
1450  9A51  20F9     JRNZ 9A4C     ;afficher blancs jusqu'a ce que compteur=0
1451  9A53  3A4C97   LD A,(974C)   ;blancs supplementaires a repartir dans A
1452  9A56  FE00     CP 00         ;= 0 ?
1453  9A58  280A     JRZ 9A64      ;oui
1454  9A5A  3D       DEC A         ;non, decrementer compteur de blcs suppl.
1455  9A5B  324C97   LD (974C),A   ;ranger...
1456  9A5E  3E20     LD A,20       ;et afficher un blanc
1457  9A60  CD5ABB   CALL BB5A     ;
1458  9A63  05       DEC B         ;decrementer compteur general
1459  9A64  3E01     LD A,01       ;rendre faux l'indicateur Z
1460  9A66  3C       INC A         ;
1461  9A67  C9       RET           ;et retour
```

```
      ** SOUS-PROGRAMME 56 **

1462  9A68  E5        PUSH HL       ;preserver registres
1463  9A69  D5        PUSH DE       ;
1464  9A6A  C5        PUSH BC       ;
1465  9A6B  3A4097    LD A,(9740)   ;numero canal dans A
1466  9A6E  FE01      CP 01         ;imprimante ?
1467  9A70  2007      JRNZ 9A79     ;non
1468  9A72  23        INC HL        ;oui, pointer HL sur 1ere valeur du c.d.c.
1469  9A73  7E        LD A,(HL)     ;valeur dans A
1470  9A74  FE31      CP 31         ;= 1 ? (code ASCII de 1=&31)
1471  9A76  CCDF9B    CALL Z,9BDF   ;si oui, executer code de controle
1472  9A79  C1        POP BC        ;recuperer registres
1473  9A7A  D1        POP DE        ;
1474  9A7B  E1        POP HL        ;
1475  9A7C  23        INC HL        ;pointer HL sur octet apres c.d.c.
1476  9A7D  23        INC HL        ;
1477  9A7E  23        INC HL        ;
1478  9A7F  23        INC HL        ;
1479  9A80  23        INC HL        ;
1480  9A81  C9        RET           ;

                  ****************************************
                  ****************************************
```

```
      ** IMPRIMER A PARTIR D'UNE PAGE SPECIFIEE **

1481  9A92  CDB29A    CALL 9AB2     ;transformer 'SORTIE IMPRIMANTE' en sp
1482  9A95  3A4197    LD A,(9741)   ;numero 1ere page dans A
1483  9A98  47        LD B,A        ;puis dans B
1484  9A99  3A4297    LD A,(9742)   ;page actuelle dans A
1485  9A9C  B8        CP B          ;page actuelle=page specifiee ?
1486  9A9D  2006      JRNZ 9AA5     ;non
1487  9A9F  CDC59A    CALL 9AC5     ;oui, restaurer 'SORTIE IMPRIMANTE'
1488  9AA2  C3DA9A    JP 9ADA       ;* VERS SORTIE IMPRIMANTE
1489  9AA5  C5        PUSH BC       ;preserver numero 1ere page
1490  9AA6  CDE09A    CALL 9AE0     ;appeler 'SORTIE IMPRIMANTE' comme un sp
1491  9AA9  C1        POP BC        ;recuperer numero 1ere page
1492  9AAA  FE01      CP 01         ;impression terminee ?
1493  9AAC  C8        RET Z         ;oui, fini
1494  9AAD  FE04      CP 04         ;fin de texte atteinte ?
1495 9AAF  C8        RET Z         ;oui, fini
1496  9AB0  18E7      JR 9A99       ;non, boucler

      ** SOUS-PROGRAMME 57 **

1497  9AB2  3EC9      LD A,C9       ;&C9 EN &9ADA (SORTIE IMPRIMANTE en sp)
1498  9AB4  21DA9A    LD HL,9ADA    ;
1499  9AB7  77        LD (HL),A     ;
1500  9AB8  218A9B    LD HL,9B8A    ;couper sp 64
1501  9ABB  77        LD (HL),A     ;
```

```
1502  9ABC  21689B    LD HL,9B68    ;couper sp 62
1503  9ABF  77        LD (HL),A     ;
1504  9AC0  21309B    LD HL,9B30    ;couper sp 59
1505  9AC3  77        LD (HL),A     ;
1506  9AC4  C9        RET           ;
```

** SOUS-PROGRAMME 58 **

```
1507  9AC5  21DA9A    LD HL,9ADA    ;restaurer 'SORTIE IMPRIMANTE'
1508  9AC8  36CD      LD (HL),CD    ;
1509  9ACA  218A9B    LD HL,9B8A    ;remettre sp 64 en service
1510  9ACD  36E5      LD (HL),E5    ;
1511  9ACF  21689B    LD HL,9B68    ;remettre sp 62 en service
1512  9AD2  36E5      LD (HL),E5    ;
1513  9AD4  21309B    LD HL,9B30    ;remettre sp 59 en service
1514  9AD7  36E5      LD (HL),E5    ;
1515  9AD9  C9        RET           ;
```

** SORTIE IMPRIMANTE **

```
1516  9ADA  CD1BBB    CALL BB1B     ;interrogation clavier
1517  9ADD  FE8B      CP 8B         ;petite touche ENTER ?
1518  9ADF  C8        RET Z         ;oui, fini
1519  9AE0  3A8E3A    LD A,(3A8E)   ;parametre 'marge haute' dans A
1520  9AE3  FE00      CP 00         ;egal a 0 ?
```

```
1521  9AE5  2809      JRZ 9AF0       ;oui
1522  9AE7  47        LD B,A         ;non, alors marge haute dans B
1523  9AE8  CD3F9B    CALL 9B3F      ;effectuer B sauts de ligne
1524  9AEB  21E59A    LD HL,9AE5     ;mettre hors service partie 'marge haute'
1525  9AEE  3618      LD (HL),18     ;
1526  9AF0  CD3D98    CALL 983D      ;calculer nbre de caracteres par ligne
1527  9AF3  CD309B    CALL 9B30      ;pose marge gauche imprimante selon (14988)
1528  9AF6  2A4597    LD HL,(9745)   ;adresse ligne actuelle dans HL
1529  9AF9  E5        PUSH HL        ;preserver
1530  9AFA  CDD298    CALL 98D2      ;calculer nbre de mots et nombre de lettres
1531  9AFD  224797    LD (9747),HL   ;addrese prochaine ligne en (38727)
1532  9B00  E1        POP HL         ;recuperer adresse ligne actuelle
1533  9B01  FE06      CP 06          ;mot trop long pour les marges ?
1534  9B03  2006      JRNZ 9B0B      ;non
1535  9B05  3E01      LD A,01        ;oui, mettre semaphore
1536  9B07  324D97    LD (974D),A    ;
1537  9B0A  C9        RET            ;fini
1538  9B0B  FE01      CP 01          ;plus rien a faire ?
1539  9B0D  C8        RET Z          ;non, alors fini
1540  9B0E  FE02      CP 02          ;simple saut de ligne ?
1541  9B10  2806      JRZ 9B18       ;oui
1542  9B12  CDDF99    CALL 99DF      ;non, sortir ligne
1543  9B15  FE04      CP 04          ;fin de texte atteinte ?
1544  9B17  C8        RET Z          ;oui, alors fini
1545  9B18  E5        PUSH HL        ;preserver HL
1546  9B19  0601      LD B,01        ;1 dans B
1547  9B1B  CD3F9B    CALL 9B3F      ;sauter B ligne
```

```
1548  9B1E  CD749B   CALL 9B74      ;si interligne=2, sauter encore 1 ligne
1549  9B21  CDAD98   CALL 98AD      ;sauter eventuelles lignes c.d.c.
1550  9B24  E1       POP HL         ;recuperer HL
1551  9B25  CDEE9B   CALL 9BEE      ;executer c.d.c. entre HL et (38727)
1552  9B28  2A4797   LD HL,(9747)  ;adresse prochaine ligne dans HL
1553  9B2B  224597   LD (9745),HL ;prochaine ligne devient ligne actuelle
1554  9B2E  18AA     JR 9ADA        ;reprendre

      ** SOUS-PROGRAMME 59 **

1555  9B30  E5       PUSH HL        ;preserver HL
1556  9B31  78       LD A,B         ;marge gauche dans A
1557  9B32  32899C   LD (9C89),A   ;ranger valeur en (40073)
1558  9B35  061E     LD B,1E        ;* EXECUTER SEQUENCE No 30
1559  9B37  CDC49B   CALL 9BC4      ;
1560  9B3A  E1       POP HL         ;recuperer HL
1561  9B3B  C9       RET            ;

      ** SOUS-PROGRAMME 60 **

1562  9B3F  E5       PUSH HL        ;preserver registres
1563  9B40  F5       PUSH AF        ;
1564  9B41  C5       PUSH BC        ;
1565  9B42  CD4E9B   CALL 9B4E      ;decrementer compteur de ligne
1566  9B45  CD689B   CALL 9B68      ;passer a la ligne suivante
```

```
1567  9B48  C1        POP BC          ;recuperer B (= nombre de lignes a sauter)
1568  9B49  10F6      DJNZ 9B41       ;si B<>0,boucler
1569  9B4B  F1        POP AF          ;recuperer registres
1570  9B4C  E1        POP HL          ;
1571  9B4D  C9        RET             ;

      ** SOUS-PROGRAMME 61 **

1572  9B4E  214497    LD HL,9744      ;HL pointe sur compteur de ligne
1573  9B51  35        DEC (HL)        ;compteur de ligne-1
1574  9B52  C0        RET NZ          ;si compteur <> 0, alors fini
1575  9B53  3A903A    LD A,(3A90)     ;restaurer compteur de ligne
1576  9B56  324497    LD (9744),A     ;
1577  9B59  21E59A    LD HL,9AE5      ;deverouiller 'marge hte' de SORTIE IMPRIM.
1578  9B5C  3628      LD (HL),28      ;
1579  9B5E  214297    LD HL,9742      ;numero de page + 1
1580  9B61  34        INC (HL)        ;
1581  9B62  3E0C      LD A,0C         ;code imprimante changement de page ds A
1582  9B64  CD8A9B    CALL 9B8A       ;executer
1583  9B67  C9        RET             ;

      ** SOUS-PROGRAMME 62 **

1584  9B68  E5        PUSH HL         ;preserver registres
1585  9B69  F5        PUSH AF         ;
```

```
1586  9B6A  C5        PUSH BC       ;
1587  9B6B  0601      LD B,01       ;sauter 1 ligne
1588  9B6D  CDC49B    CALL 9BC4     ;
1589  9B70  C1        POP BC        ;recuperer registres
1590  9B71  F1        POP AF        ;
1591  9B72  E1        POP HL        ;
1592  9B73  C9        RET           ;

      ** SOUS-PROGRAMME 63 **

1593  9B74  F5        PUSH AF       ;preserver AF
1594  9B75  3A8F3A    LD A,(3A8F)   ;parametre 'interligne  dans A
1595  9B78  FE02      CP 02         ;= 2 ?
1596  9B7A  2802      JRZ 9B7E      ;oui
1597  9B7C  F1        POP AF        ;non, recuperer A et retour
1598  9B7D  C9        RET           ;
1599  9B7E  0601      LD B,01       ;oui, sauter 1 ligne
1600  9B80  CD3F9B    CALL 9B3F     ;
1601  9B83  F1        POP AF        ;recuperer A et retour
1602  9B84  C9        RET           ;

      ** SOUS-PROGRAMME 64 **

1603  9B8A  E5        PUSH HL       ;preserver registres
1604  9B8B  C5        PUSH BC       ;
```

```
1605  9B8C  FEF4      CP F4          ;caractere redefini ? (244-254)
1606  9B8E  3809      JRC 9B99       ;non
1607  9B90  D6F2      SUB F2         ;oui, No sequence corespondante dans A
1608  9B92  47        LD B,A         ;puis dans B
1609  9B93  CDC49B    CALL 9BC4      ;executer sequence No B
1610  9B96  C1        POP BC         ;recuperer registres
1611  9B97  E1        POP HL         ;
1612  9B98  C9        RET            ;fini
1613  9B99  060D      LD B,0D        ;B pret pour sequence 13
1614  9B9B  FE40      CP 40          ;arobas ?
1615  9B9D  2814      JRZ 9BB3       ;oui
1616  9B9F  04        INC B          ;non, B pret pour sequence 14
1617  9BA0  FE5B      CP 5B          ;crochet gauche ?
1618  9BA2  280F      JRZ 9BB3       ;oui
1619  9BA4  04        INC B          ;non, B pret pour sequence 15
1620  9BA5  FE5D      CP 5D          ;crochet droit ?
1621  9BA7  280A      JRZ 9BB3       ;oui
1622  9BA9  04        INC B          ;non, B pret pour sequence 16
1623  9BAA  FE7B      CP 7B          ;parenthese gauche ?
1624  9BAC  2805      JRZ 9BB3       ;oui
1625  9BAE  04        INC B          ;non, B pret pour sequence 17
1626  9BAF  FE7D      CP 7D          ;parenthese droite ?
1627  9BB1  2006      JRNZ 9BB9      ;non, alors caractere normal
1628  9BB3  CDC49B    CALL 9BC4      ;executer sequence B
1629  9BB6  C1        POP BC         ;recuperer registres
1630  9BB7  E1        POP HL         ;
1631  9BB8  C9        RET            ;fini
```

```
1632  9BB9  CD2EBD   CALL BD2E     ;imprimante disponible ?
1633  9BBC  38FB     JRC 9BB9      ;boucler jusqu'a disponibilite
1634  9BBE  CD31BD   CALL BD31     ;sortir caractere
1635  9BC1  C1       POP BC        ;recuperer registres
1636  9BC2  E1       POP HL        ;
1637  9BC3  C9       RET           ;
```

** SOUS-PROGRAMME 65 **

```
1638  9BC4  210E9C   LD HL,9C0E    ;HL pointe sur debut table des sequences-1
1639  9BC7  3EFF     LD A,FF       ;code de separation dans A
1640  9BC9  23       INC HL        ;HL pointe sur adresse suivante
1641  9BCA  BE       CP (HL)       ;code de separation ?
1642  9BCB  20FC     JRNZ 9BC9     ;non, continuer a chercher
1643  9BCD  05       DEC B         ;separat. trouve / No sequence correspond ?
1644  9BCE  20F9     JRNZ 9BC9     ;non, continuer a chercher
1645  9BD0  23       INC HL        ;sequence trouvee/HL pointe sur 1ere valeur
1646  9BD1  7E       LD A,(HL)     ;valeur dans A
1647  9BD2  FEFF     CP FF         ;fin de sequence ?
1648  9BD4  C8       RET Z         ;oui, fini
1649  9BD5  CD2EBD   CALL BD2E     ;non, imprimante disponible ?
1650  9BD8  38FB     JRC 9BD5      ;
1651  9BDA  CD31BD   CALL BD31     ;sortir caract. ou executer c.d.c.
1652  9BDD  18F1     JR 9BD0       ;continuer sequence
```

```
      ** SOUS-PROGRAMME 66 **

1653  9BDF  23        INC HL        ;HL pointe sur 2eme octet code de ctrl
1654  9BE0  7E        LD A,(HL)     ;dans A
1655  9BE1  D61E      SUB 1E        ;code ASCII devient valeur reelle
1656  9BE3  FE1E      CP 1E         ;valeur < 30 ?
1657  9BE5  D0        RET NC        ;non, donc valeur illegale / retour
1658  9BE6  FE12      CP 12         ;valeur >= 18 ?
1659  9BE8  D8        RET C         ;non, donc valeur illegale / retour
1660  9BE9  47        LD B,A        ;No de sequence dans B
1661  9BEA  CDC49B    CALL 9BC4     ;executer
1662  9BED  C9        RET           ;

      ** SOUS-PROGRAMME 67 **

1663  9BEE  ED5B4797  LD DE,(9747)  ;adresse prochaine ligne dans DE
1664  9BF2  CDED95    CALL 95ED     ;comparer HL et DE
1665  9BF5  FEFF      CP FF         ;HL < DE ?
1666  9BF7  C0        RET NZ        ;non, fini
1667  9BF8  7E        LD A,(HL)     ;HL pointe sur code de controle ?
1668  9BF9  FEF1      CP F1         ;
1669  9BFB  2803      JRZ 9C00      ;oui
1670  9BFD  23        INC HL        ;non, pointer HL sur adresse suivante...
1671  9BFE  18F2      JR 9BF2       ;...et continuer
1672  9C00  CD689A    CALL 9A68     ;executer code de controle...
1673  9C03  18ED      JR 9BF2       ;...et continuer
```

```
** TABLE DES SEQUENCES IMPRIMANTE **

9C0F  FF                        ;separateur
9C10  A                         ;* SEQUENCE 1: A LA LIGNE
9C11  D                         ;
9C12  FF                        ;separateur
9C13  6F                        ;* SEQUENCE 2: o + ACCENT CIRCONFLEXE
9C14  8                         ;
9C15  5E                        ;
9C16  FF                        ;separateur
9C17  61                        ;* SEQUENCE 3: a + ACCENT CIRCONFLEXE
9C18  8                         ;
9C19  5E                        ;
9C1A  FF                        ;separateur
9C1B  69                        ;* SEQUENCE 4: i + DOUBLE POINT
9C1C  8                         ;
9C1D  7E                        ;
9C1E  FF                        ;separateur
9C1F  75                        ;* SEQUENCE 5: u + ACCENT CIRCONFLEXE
9C20  8                         ;
9C21  5E                        ;
9C22  FF                        ;separateur
9C23  40                        ;* SEQUENCE 6: a + ACCENT GRAVE
9C24  FF                        ;separateur
9C25  5C                        ;* SEQUENCE 7: c + CEDILLE
9C26  FF                        ;separateur
9C27  7C                        ;* SEQUENCE 8: u + ACCENT GRAVE
```

```
9C28  FF                    ;separateur
9C29  7B                    ;* SEQUENCE 9:  e + ACCENT AIGU
9C2A  FF                    ;separateur
9C2B  7D                    ;* SEQUENCE 10: e + ACCENT AIGU
9C2C  FF                    ;separateur
9C2D  65                    ;* SEQUENCE 11: e + ACCENT CIRCONFLEXE
9C2E  8                     ;
9C2F  5E                    ;
9C30  FF                    ;separateur
9C31  69                    ;* SEQUENCE 12: i + ACCENT CIRCONFLEXE
9C32  8                     ;
9C33  5E                    ;
9C34  FF                    ;separateur
9C35  1B                    ;* SEQUENCE 13: AROBAS
9C36  52                    ;
9C37  0                     ;
9C38  40                    ;
9C39  1B                    ;
9C3A  52                    ;
9C3B  1                     ;
9C3C  FF                    ;separateur
9C3D  1B                    ;* SEQUENCE 14: CROCHET GAUCHE
9C3E  52                    ;
9C3F  0                     ;
9C40  5B                    ;
9C41  1B                    ;
9C42  52                    ;
```

```
9C43  1                         ;
9C44  FF                        ;separateur
9C45  1B                        ;* CROCHET DROIT
9C46  52                        ;
9C47  Ø                         ;
9C48  5D                        ;
9C49  1B                        ;
9C4A  52                        ;
9C4B  1                         ;
9C4C  FF                        ;separateur
9C4D  1B                        ;* SEQUENCE 16: PARENTHESE GAUCHE
9C4E  52                        ;
9C4F  Ø                         ;
9C5Ø  7B                        ;
9C51  1B                        ;
9C52  52                        ;
9C53  1                         ;
9C54  FF                        ;separateur
9C55  1B                        ;* SEQUENCE 17: PARENTHESE DROITE
9C56  52                        ;
9C57  Ø                         ;
9C58  7D                        ;
9C59  1B                        ;
9C5A  52                        ;
9C5B  1                         ;
9C5C  FF                        ;separateur
9C5D  1B                        ;* SEQUENCE 18: METTRE NLQ
```

```
9C5E  78                ;
9C5F  1                 ;
9C60  FF                ;separateur
9C61  1B                ;* SEQUENCE 19: ENLEVER NLQ
9C62  78                ;
9C63  0                 ;
9C64  FF                ;separateur
9C65  1B                ;* SEQUENCE 20: METTRE ITALIQUE
9C66  34                ;
9C67  FF                ;separateur
9C68  1B                ;* SEQUENCE 21: ENLEVER ITALIQUE
9C69  35                ;
9C6A  FF                ;separateur
9C6B  1B                ;* SEQUENCE 22: METTRE CARACTERES GRAS
9C6C  45                ;
9C6D  FF                ;separateur
9C6E  1B                ;* SEQUENCE 23: ENLEVER CARACTERES GRAS
9C6F  46                ;
9C70  FF                ;separateur
9C71  1B                ;* SEQUENCE 24: METTRE SOULIGNEMENT
9C72  2D                ;
9C73  1                 ;
9C74  FF                ;separateur
9C75  1B                ;* SEQUENCE 25: ENLEVER SOULIGNEMENT
9C76  2D                ;
9C77  0                 ;
9C78  FF                ;separateur
```

```
9C79  1B                ;* SEQUENCE 26: METTRE CARACTERES MINI
9C7A  4D                ;
9C7B  FF                ;separateur
9C7C  1B                ;* SEQUENCE 27: ENLEVER CARACTERES MINI
9C7D  50                ;
9C7E  FF                ;separateur
9C7F  1B                ;* SEQUENCE 28: METTRE DOUBLE LARGEUR
9C80  57                ;
9C81  1                 ;
9C82  FF                ;separateur
9C83  1B                ;* SEQUENCE 29: ENLEVER DOUBLE LARGEUR
9C84  57                ;
9C85  0                 ;
9C86  FF                ;separateur
9C87  1B                ;* SEQUENCE 30: POSE MARGE GAUCHE
9C88  6C                ;
9C89  0                 ;valeur definie en cours de programme
9C8A  FF                ;* FIN DE LA TABLE
```

■ LES VARIABLES

Deux blocs de variables sont implantés, le premier étant utilisé par la partie du programme s'occupant de la gestion de l'écran de travail, et le second par les parties *Sortie écran* et *Sortie imprimante.*

PREMIER BLOC (DE 14976 À 14992)

14976 (&3A80)

Placard de rangement provisoire de différentes valeurs à différents endroits du programme.

14977 (&3A81)

Idem, et sert d'autre part à préserver provisoirement le caractère qui vient d'être saisi au clavier, lorsque les registres sont utilisés pour autre chose.

14978 et 14979 (&3A82 et &3A83)

Placard de rangement provisoire de différentes valeurs supérieures à 255 et tenant donc sur deux octets.

14980 (&3A84)

Ligne écran actuelle du curseur. Cette ligne écran ne doit pas être confondue avec la ligne théorique (même si un texte fait 100 ou 200 lignes, le curseur écran sera toujours compris entre 2 et 23 inclus, les lignes 1, 24 et 25 n'étant pas utilisées par l'écran de travail).

14981 (&3A85)

Ligne théorique actuelle du curseur. Il s'agit cette fois du numéro de ligne du texte. L'écran de travail entièrement " déroulé " comprend 255 lignes de 80 colonnes (ce qui correspond bien à 80*255 = 20 400 caractères). En termes de texte justifié, il est bien évident que le nombre de lignes peut être

bien plus important ; tout dépend, à ce moment-là, des marges, des sauts de ligne, etc.

14982 (&3A86)

Colonne actuelle du curseur (entre 1 et 80 inclus).

14983 (&3A 87)

Sémaphore d'insertion (mis à 1 si le mode insertion est demandé, mis à 0 autrement).

14984 et 14985 (&3A88 et &3A89)

Curseur mémoire. La valeur contenue dans ce placard évolue d'une manière rigoureusement parallèle à la position du curseur écran. En d'autres termes, tout caractère pointé sur l'écran par le curseur est également pointé en RAM par le curseur mémoire (si le curseur écran avance d'une position, le curseur mémoire fait de même ; si le curseur écran descend d'une ligne, le curseur mémoire avance de 80, etc.).

14986 et 14987 (&3A8B et &3A8B)

Pointeur de fin de texte. Cette variable représente l'adresse du dernier caractère du texte. (Notons que ce caractère peut tout aussi bien être un espace inutile, symbolisé sur l'écran par un point). Si, par exemple, le curseur écran est descendu d'une ligne alors qu'il se trouvait en fin de texte, le pointeur de fin de texte sera augmenté de 80, et les 80 derniers caractères du texte ne seront que des espaces inutiles.

Rappelons que la RAM de stockage du texte va de 15000 à 35399.

Les variables suivantes sont celles définissant le format (se reporter éventuellement au mode d'emploi) :

14988 (&3A8C) Marge gauche.

14989 (&3A8D) Marge droite.

14990 (&3A8E) Marge haute.

14991 (&3A8F) Interligne.

14992 (&3A90) Nombre de lignes par page.

DEUXIÈME BLOC : 38720 À 38733 (&9740 À &974D)

38720 (&9740)

Numéro du canal de sortie du texte (0 pour une sortie écran et 1 pour une sortie imprimante).

38721 (9741)

Numéro de la page à partir de laquelle le texte doit être affiché sur 1 écran ou imprimé.

38722 (&9742)

Numéro de la page actuellement traitée par le programme de justification.

38723 (&9743)

Chaque fois qu'une ligne est traitée par le programme de justification, les codes de contrôle indiquant un ou des sauts de lignes sont repérés, et le nombre de lignes(s) correspondant est rangé dans ce placard (ou additionné à celui qui s'y trouve déjà). Lorsque la ligne est affichée ou imprimée, les sauts sont effectués et le placard est remis à 0.

37824 (&9744)

Sert de compteur pour que chaque page affichée ou imprimée comporte le nombre de lignes souhaitée (la valeur initiale est copiée dans le placard 14992).

38725 et 38726 (&9745 et &9746)

Adresse RAM du début de la ligne actuellement traitée par le programme de justification.

38727 et 38728 (&9747 et &9748)

Adresse RAM du début de la prochaine ligne qui devra être traitée par le programme de justification.

Concernant ces deux placards, il faut préciser qu'en réalité, le terme deligne est quelque peu impropre. En effet, le principe de sortie d'un texte justifié est très schématiquement le suivant : à partir d'un caractère rangé en RAM, avancer mot par

mot jusqu'à ce que "l'on" se rende compte qu'un mot ne rentrera plus dans la ligne (le tout, bien sûr, en tenant compte de tout ce qui ne doit pas apparaître, comme les codes de contrôle, par exemple). A ce moment-là seulement, on peut considérer que la ligne actuelle est bornée d'une manière précise (et donc en profiter pour ranger dans le placard 38727 ce qui est en fait l'adresse du début de la zone RAM d'où sera tirée la ligne suivante).

38729 (&9749)

Numéro de la ligne écran où doit s'afficher la ligne de texte actuellement traitée par le programme de justification (il faut bien sûr repérer le moment où la ligne 23 est atteinte et, le cas échéant, faire un *scrolling* d'écran).

38730 (&974A)

Nombre de caractères par ligne (cette variable dépend naturellement des marges qui ont été définies par l'utilisateur).

38731 (&974B)

Nombre de blancs à mettre entre chaque mot pour obtenir une ligne justifiée.

38732 (&974C)

Nombre de blancs supplémentaires à répartir pour obtenir une ligne justifiée.

Un exemple permettra sans doute une meilleure compréhension du rôle de ces deux variables.

Imaginons que l'utilisateur ait positionné la marge gauche en 10 et la marge droite en 60. Le nombre de caractères par ligne est donc 60—10+1=51. Le résultat des calculs de justification d'une ligne donnée fait apparaître qu'il est possible de faire entrer par exemple 9 mots dont la somme totale des lettres qui les composent égale 37. Le nombre maximal de caractères étant 51, il restera 51—37=14 espaces blancs à répartir dans cette ligne. Il faut donc commencer par effectuer la division suivante : 14/8 (n'oublions pas que 9 mots impliquent 8 intervalles), le résultat étant 1 et le reste 6. La valeur 1 sera donc rangée dans le placard 38731, et la valeur 6 dans le placard 38732.

Il faut bien comprendre que ces 6 blancs supplémentaires devront impérativement être utilisés si l'on veut que la ligne comporte exactement 51 caractères ; en l'occurrence, il faudra les répartir entre les 7 premiers mots.

Pour résumer, notre ligne aura donc l'aspect suivant :

1er mot
2 blancs
2e mot
2 blancs
3e mot
2 blancs
4e mot
2 blancs
5e mot
2 blancs
6e mot
2 blancs
7e mot
1 blanc
8e mot
1 blanc
9e mot

38733 (&974D)

Sémaphore indiquant, s'il est mis à 1, que le programme a rencontré un mot trop long pour tenir dans les marges spécifiées, et que la justification est par conséquent impossible.

■ LES SOUS-PROGRAMMES

Sous-programme 1

Appelé lorsque le curseur écran doit être déplacé de 1 vers la droite (simple déplacement ou saisie d'un caractère), il est chargé de modifier les variables en conséquence (colonne, curseur mémoire, et éventuellement le pointeur de fin de texte, la ligne écran et la ligne théorique). Un éventuel dépassement de la ligne 23 est naturellement testé, et le cas échéant traité.

Registres d'entrée : Indifférents.

Registres de sortie : AF et DE sont modifiés.

Sous-programme 2

Idem que le précédent, mais concernant le déplacement du curseur vers le bas (passage d'une L à une ligne L + 1).

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont modifiés sauf BC.

Sous-programme 3

Appelé après un scrolling vers le haut (la ligne 2 disparaît et les lignes 3 à 23 deviennent les lignes 2 à 22), il est chargé d'afficher la nouvelle ligne 23 d'après la RAM. A noter que ce sous-programme est appelé avant toute modification des variables.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont modifiés.

Sous-programme 4

Idem que le sous-programme 1, mais concernant le déplacement du curseur vers la gauche. Un éventuel dépassement par le bas de la ligne écran n° 2 est testé et, le cas échéant, traité.

Registres d'entrée : Indifférents.

Registres d'appel : Tous les registres sont modifiés.

Sous-programme 5

Idem, mais concernant le déplacement du curseur vers le

haut. Là encore, un éventuel dépassement par le bas de la ligne 2 est testé.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont modifiés.

Sous-programme 6

Appelé après un scrolling vers le bas (la ligne 23 disparaît et les lignes 2 à 22 deviennent les lignes 3 à 23), il est chargé d'afficher la nouvelle ligne 2 d'après la RAM. Ce sous-programme est appelé avant toute modification des variables.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont modifiés.

Sous-programme 7

Après que l'adresse RAM du début de la dernière page de l'écran de travail a été trouvée (adresse qui doit être un multiple de 1 760 puisque l'écran peut contenir 22*80=1 760 caractères), ce sous-programme calcule la valeur des différentes variables (ligne écran, ligne théorique, colonne et curseur mémoire) pour qu'elles correspondent au dernier caractère de cette dernière page (ce sous-programme est utilisé par la commande SHIFT + CURSEUR BAS).

Registres d'entrée : HL contient l'adresse RAM du premier caractère de la dernière page (n'oublions pas que cette dernière page n'est pas forcément complète), et A contient le numéro de la ligne théorique correspondant à ce premier caractère.

Registres de sortie : HL est préservé.

Sous-programme 8 :

Division de HL par BC.

Registres de sortie : Résultat dans A, reste dans HL, BC est intact.

Sous-programme 9

A partir d'une adresse RAM spécifiée, rechercher celle du premier code d'espace inutile (255/&FF) ou du premier code

de fin de paragraphe (242/&F2). Si aucun de ces deux codes n'est rencontré avant la fin du texte, c'est l'adresse RAM de la fin de texte qui est ramenée.

Registres d'entrée : HL contient l'adresse à partir de laquelle doit s'effectuer la recherche.

Registres de sortie : HL est préservé, DE contient l'adresse recherchée et A sert d'indicateur (1 si un espace inutile a été rencontré en premier, 2 s'il s'agit d'une fin de paragraphe, et 3 s'il s'agit de la fin du texte).

Sous-programme 10

Ecrire B espaces inutiles en RAM à partir de l'adresse contenue dans DE.

Registres de sortie : BC et AF sont modifiés.

Sous-programme 11

Nettoyer les 80 adresses (donc y écrire le code 255) situées aprés 35399.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont préservés.

Sous-programme 12

Décaler la zone RAM de HL à DE inclus de + A.

Registres de sortie : HL et DE sont préservés.

Sous-programme 13

Décaler la zone RAM de HL à DE inclus de — A.

Registres de sortie : HL et DE sont préservés.

Sous-programme 14

Afficher sur l'écran, et à partir de la position actuelle du curseur, la zone RAM de HL à DE inclus (si la zone spécifiée est trop longue pour être affichée en entier sur l'écran, le sous-programme n'affiche que ce qui est possible).

Registres d'entrée : L'adresse spécifiée par HL doit impérativement correspondre à la position du curseur sur l'écran.

Registres de sortie : HL et DE sont préservés.

Sous-programme 15

Afficher les DE caractères rangés en RAM à partir de l'adresse contenue dans HL (cette adresse étant incluse).

Registres d'entrée : BC doit contenir l'adresse écran à partir de laquelle se fera l'affichage.

Registres de sortie : Tous les registres sont modifiés.

Sous-programme 16

Sortir un caractère normal d'après la table du générateur de caractères, ou un caractère redéfini d'après la table de redéfinition (voir explications complémentaires au paragraphe traitant des points particuliers).

Registres d'entrée : BC contient l'adresse écran où doit s'afficher le caractère, et A contient le code ASCII de ce caractère.

Registres de sortie : Seul BC est inchangé.

Sous-programme 17

Teste si le curseur mémoire est à la position minimale (15000). Si oui, l'indicateur Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF, HL et DE sont modifiés.

Sous programme 18

Teste si le curseur mémoire est à la position maximale (35399). Si oui, l'indicateur Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF, HL et DE sont modifiés (ce dernier contient l'adresse du curseur mémoire).

Sous-programme 19

Teste si le curseur mémoire est égal au pointeur de fin de texte. Si oui, l'indicateur Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF et DE sont modifiés (ce dernier contient l'adresse du curseur mémoire).

Sous-programme 20

Ajoute au curseur mémoire le contenu de A.

Registres de sortie : Tous les registres sont préservés.

Sous-programme 21

Soustrait du curseur mémoire le contenu de A.

Registres de sortie : AF est modifié.

Sous-programme 22

Teste si le pointeur de fin de texte est égal au maximum (35399). Si oui, Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 23

Additionne au pointeur de fin de texte le contenu de A.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont préservés.

Sous-programme 24

Soustrait du pointeur de fin de texte le contenu de A.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 25

Teste si la ligne théorique actuelle (celle où se trouve le curseur) est égale au maximum (255). Si oui, Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 26

Teste si la ligne écran actuelle est égale à 2. Si oui, Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 27

Teste si la ligne écran actuelle est égale à 23. Si oui, Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 28

Teste si la colonne actuelle est égale à 80. Si oui, Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 29

Teste si la colonne actuelle est égale à 1. Si oui, Z est vrai au retour.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 30

Positionne le curseur système à l'emplacement spécifié par les variables (14982) et (14980).

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont préservés.

Sous-programme 31

Affiche, à la position actuelle du curseur écran, le caractère adressé par le curseur mémoire.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 32

Affiche un curseur à la position indiquée par les variables (14982) et (14980).

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 33

Scrolling vers le haut des lignes écran 3 à 23 incluses.

Registres d'entrée : Indifférents

Registres de sortie : Tous les registres sont modifiés.

Sous-programme 34

Scrolling vers le bas des lignes écran 2 à 22 incluses.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont modifiés.

Sous-programme 35

Compare les valeurs contenues dans les registres HL et DE.

Registres de sortie : HL, DE et BC sont préservés. A contient 1 si HL > DE, 0 si HL = DE et 255 si HL < DE.

Sous-programme 36

Utilisé chaque fois que le curseur est déplacé, il affiche en bas et à gauche de l'écran la nouvelle ligne et/ou la nouvelle colonne (voir explications complémentaires au paragraphe traitant des points particuliers).

Registres d'entrée : Si A contient 0, seule la colonne sera affichée ; si A contient 2, seule la ligne sera affichée ; si A contient 1, les deux le seront.

Registres de sortie : HL et AF sont modifiés.

Sous-programme 37

Nettoie la zone RAM de HL inclus à 39400 inclus.

Registres de sortie : AF est modifié.

Sous-programme 38

Modifie les variables pour qu'elles correspondent à un positionnement du curseur mémoire à l'adresse 15000 (et donc à un positionnement du curseur écran en ligne théorique 1, ligne écran 2 et colonne 1).

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont préservés.

Sous-programme 39

Coupe l'accès au sous-programme 44, transforme d'autre part la partie SORTIE ÉCRAN (lignes 1094 à 1130 du listing assembleur) en sous-programme, de manière qu'il puisse être appelé par un CALL &97E1, et supprime enfin tous les appels d'affichage (routine système &BB5A) des sous-programmes 45 et 55. Ce sous-programme est utilisé lorsque l'on souhaite une sortie écran à partir d'une page spécifiée : jusqu'à ce que la page soit atteinte, les calculs de justification sont effectués, mais rien n'apparaît sur l'écran.

Registres d'entrée : Indifférents.

Registres de sortie : HL et AF sont modifiés.

Sous-programme 40

Remet en l'état tout ce qui a été modifié par le sous-programme précédent.

Registres d'entrée : Indifférents.
Registres de sortie : HL et AF sont modifiés.

Sous-programme 41

Positionne le curseur système en fonction de la marge gauche (14988) et de la ligne (38729).

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

Sous-programme 42

Calcule, en fonction des marges, le nombre de caractères par ligne. Le résultat est rangé en (38730).

Registres d'entrée : Indifférents.

Registres de sortie : AF et BC sont modifiés.

Sous-programme 43

Effectue B sauts de ligne, en vérifiant au fur et à mesure le compteur de lignes.

Registres de sortie : HL et AF son préservés.

Sous-programme 44

Passe à la ligne suivante (en incrémentant le compteur de lignes). Tout dépassement éventuel de la ligne 23 est vérifié et traité le cas échéant.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont préservés.

Sous-programme 45 :

Décrémente de 1 le compteur de lignes. Si ce compteur = 0, une ligne en pointillé est affichée et le compteur de pages est incrémenté.

Registres d'entrée : Indifférents.

Registres de sortie : Seul DE est préservé.

Sous-programme 46

Vérifie la valeur du paramètre " interligne ". S'il est égal à 2, une ligne est sautée.

Registres d'entrée : Indifférents.

Registres de sortie : AF et HL sont préservés.

Sous-programme 47

Après l'affichage d'une ligne, il saute le nombre de lignes indiqué par (38723) puis remet ce dernier à 0. Il s'agit là, bien sûr, des sauts de lignes relatifs aux codes de contrôle.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont modifiés.

Sous-programme 48

Pour la prochaine ligne à afficher, calculer le nombre de mots ainsi que le nombre total de lettres qui composent ces mots.

Registres d'entrée : HL pointe sur le début de la zone RAM correspondante (le nombre de caractères par ligne est d'autre part rangé dans le placard 38730).

Registres de sortie : HL pointe sur le début de la zone RAM correspondant à la prochaine ligne ; B contient le nombre de mots ; C contient le nombre de lettres ; A sert d'indicateur: il contient 1 s'il n'y a plus rien à faire (fin de texte rencontrée et B=0), 2 s'il suffit de passer une ligne (rencontre du code de contrôle de fin de paragraphe seul sur une ligne), 3 si la ligne se termine par le code de fin de paragraphe, 4 si la ligne se termine par la fin du texte, et 6 si un mot trop long pour les marges a été rencontré.

Sous-programme 49

Appelé par le précédent, ce sous-programme effectue, au fur et à mesure qu'il les rencontre, les codes de contrôle de pose de marge et de saut de ligne (les poses de marge sont effectuées immédiatement, tandis que les sauts de lignes sont "retenus" et effectués lorsque l'affichage de la ligne est terminé).

Registres d'entrée : HL pointe sur le premier des 5 octets (&F1) constituant le code de contrôle.

Registres de sortie : HL pointe sur l'octet suivant immédiatement le dernier octet du code de contrôle, et tous les autres registres sont préservés.

Sous-programme 50

Vérifie si le mot pointé par HL peut être ajouté dans la ligne en cours de justification sans que le nombre de caractères soit dépassé.

Registres d'entrée : HL pointe sur le premier caractère du mot.

Registres de sortie : Si le mot est trop long pour être ajouté dans la ligne, A = 0; sinon, A = 1. De toute façon, HL pointe sur l'octet suivant immédiatement le dernier caractère du mot (sauf dans le cas particulier où le mot est immédiatement suivi d'un code de fin de paragraphe : dans ce cas, HL pointe sur l'octet suivant immédiatement ce code).

Sous-programme 51

Calcule, en fonction du nombre de lettres que peut contenir une ligne, du nombre de mots qui peuvent entrer dans cette ligne et du nombre total des lettres qui les composent, le nombre d'espaces qui devront être mis entre chaque mot et le nombre de blancs supplémentaires à répartir (revoir éventuellement les explications concernant les variables 38731 et 38732).

Registres d'entrée : B contient le nombre de mots et C contient le nombre de lettres.

Registres de sortie : HL est préservé (le nombre de blancs entre chaque mot est rangé en (38730) et le nombre de blancs supplémentaires en (38732)).

Sous-programme 52

Teste si la valeur contenue dans HL est plus grande ou égale au pointeur de fin de texte. Si oui, A = 0 ou 1 ; sinon A = 255.

Registres de sortie : Seul AF est modifié.

Sous-programme 53

Selon le signal contenu dans A après le retour du sous-programme 48, ce sous-programme sort une ligne, saute simplement une ligne, ou ne fait rien.

Registres d'entrée : Registres de sortie du sous-programme 48.

Registres de sortie : HL pointe sur le début de la zone RAM d'où sera tirée la prochaine ligne, et AF est préservé.

Sous-programme 54

Affiche une ligne à l'écran.

Registres d'entrée : HL pointe sur le début de la ligne.

Registres de sortie : HL pointe sur le début de la ligne suivante.

Sous-programme 55

Complémentaire du précédent, il est appelé lorsqu'un mot est trouvé et affiche toutes les lettres de ce mot (rappelons que la fin d'un mot est signalée par la rencontre d'un espace inutile, d'un code de fin de paragraphe, d'un espace blanc ou d'un code de contrôle).

Registres d'entrée : HL pointe sur le premier caractère du mot, A contient ce caractère, et B contient le nombre de caractères par ligne.

Registres de sortie : HL pointe sur l'octet suivant immédiatement le dernier caractère du mot. Si ce mot était le dernier de la ligne, le compteur B a atteint 0 et l'indicateur Z est vrai au retour.

Sous-programme 56

Appelé par le sous-programme 54 chaque fois qu'un code de contrôle est rencontré, ce sous-programme vérifie s'il s'agit d'un code de contrôle imprimante et, si oui, l'exécute.

Registres d'entrée : HL pointe sur le premier des 5 octets constituant le code de contrôle.

Registres de sortie : AF est modifié et HL pointe sur l'octet suivant immédiatement le dernier des 5 octets du code de contrôle.

Sous-programme 57

Transforme la partie SORTIE IMPRIMANTE en sous-programme et coupe l'accès aux sous-programmes 59, 62 et 64. Il est utilisé pour empêcher l'impression des pages situées avant celle qui a été spécifiée lors de la demande de sortie imprimante (voir sous-programme 39 dont il est la réplique).

Registres d'entrée : Indifférents.

Registres de sortie : HL et AF sont modifiés.

Sous-programme 58

Inverse du précédent.

Registres d'entrée : Indifférents.

Registres de sortie : HL et AF sont modifiés.

Sous-programme 59

Pose de la marge gauche de l'imprimante en fonction de (14988).

Registres d'entrée : Ce sous-programme est appelé immédiatement après le sous-programme 42, au retour duquel B contient (14988).

Registres de sortie : Tous les registres sont modifiés.

Sous-programme 60

Effectue B saut(s) de ligne imprimante (en vérifiant chaque fois le compteur de lignes et le compteur de pages).

Registres de sortie : BC est modifié.

Sous-programme 61

Décrémente le compteur de lignes par page. Si ce compteur atteint 0, il est restauré, la partie "marge haute" de la SORTIE IMPRIMANTE est déverrouillée, et le compteur de pages est incrémenté.

Registres d'entrée : Indifférents.

Registres de sortie : HL et AF sont modifiés.

Sous-programme 62

Saute une ligne imprimante.

Registres d'entrée : Indifférents.

Registres de sortie : Tous les registres sont préservés.

Sous-programme 63

Vérifie si le paramètre interligne (14991) est égal à 2 et, si oui, saute une ligne imprimante.

Registres d'entrée : Indifférents.

Registres de sortie : HL et AF sont préservés.

Sous-programme 64

Sort sur l'imprimante le caractère contenu dans A, y compris s'il s'agit de {, }, [,], ou d'un caractère accentué.

Registres de sortie : AF est modifié.

Sous-programme 65

Exécute la séquence imprimante n° B (voir explications sur les séquences imprimante au paragraphe traitant des points particuliers).

Registres de sortie : HL, AF et BC sont modifiés.

Sous-programme 66

Exécute un code de contrôle imprimante.

Registres d'entrée : HL pointe sur le deuxième des 5 octets constituant le code de contrôle.

Registres de sortie : HL, BC et AF sont modifiés.

Sous-programme 67

Recherche et exécute tous les codes de contrôle imprimante de la zone mémoire comprise entre HL (exclu), et (38727) inclus (cela pour éviter que des codes de contrôle situés entre la dernière lettre du dernier mot de la ligne qui vient d'être imprimée et le début de la zone RAM d'où sera tirée la prochaine ligne ne soient laissés pour compte).

Registres de sortie : Tous les registres sont modifiés.

En plus de ces sous-programmes, un certain nombre de routines système sont appelées à différents endroits. Pour ceux qui n'auraient pas les renseignements nécessaires dans leur bibliothèque, voici la description de celles qui sont utilisées dans notre programme.

&BB1B

Teste si une touche est pressée. Si une touche est pressée, le le sémaphore de *carry* est vrai en sortie et A contient le caractère. Sinon, le sémaphore de carry est faux.

&BB2D

Définition du code qui sera fourni lors de la pression d'une touche conjointement à la touche SHIFT.

Registres d'entrée : A contient le numéro de la touche et B contient le code désiré.

Registres de sortie : HL et AF sont modifiés.

&BB5A

Affichage d'un caractère à la position actuelle du curseur système.

Registres d'entrée : A contient le caractère (s'il s'agit d'un code de contrôle, donc compris entre 0 et &1F, ce code est exécuté).

Registres de sortie : Tous les registres sont préservés.

&BB75

Positionnement du curseur système.

Registres d'entrée : H contient le numéro de colonne et L le numéro de ligne.

Registres de sortie : HL et AF sont modifiés.

&BB8A

Affichage d'un curseur sur l'écran, à la position actuelle du curseur système.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

&BC1A

Calcul de l'adresse écran d'une position du mode texte (rappelons qu'en mode 2, une position occupe 8 octets ; dans ce mode, l'adresse fournie sera celle de l'octet du haut).

Registres d'entrée : H contient la colonne et L contient la ligne. (Attention : la colonne et la ligne doivent être spécifiées sur la base d'une numérotation de 0 à 39 et de 0 à 24 ; pour obtenir par exemple l'adresse de la position (3,5), il faudrait charger H avec 2 et L avec 4.)

Registres de sortie : HL contient l'adresse, BC et AF sont modifiés.

&BC50

Déplacement d'une partie de l'écran d'une ligne vers le haut ou d'une ligne vers le bas (*scrolling*).

Registres d'entrée : B contient 0 pour un déplacement vers le bas, toute autre valeur provoquant un déplacement vers le haut. La zone à déplacer doit être délimitée comme suit :

H contient la colonne de gauche.

L contient la ligne du haut.

D contient la colonne de droite.

E contient la ligne du bas.

&BD2E

Teste si l'imprimante est disponible. Au retour, l'indicateur de carry est vrai si l'imprimante est indisponible, il est faux autrement.

&BD31

Envoie un caractère vers l'imprimante (qui doit être disponible).

Registres d'entrée : A contient le caractère.

Registres de sortie : AF est modifié (en réalité, seul F est modifié).

&B906

Sélectionne la ROM inférieure.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

&B909

Coupe la ROM inférieure et resélectionne la RAM.

Registres d'entrée : Indifférents.

Registres de sortie : AF est modifié.

■ LES POINTS PARTICULIERS

L'AFFICHAGE DES CARACTÈRES

Lors des saisies de caractères (lorsque vous frappez votre texte), il ne se pose pas de problème particulier concernant la vitesse d'exécution de l'affichage, et le programme utilise tout simplement la routine système &BB5A évoquée plus haut. Même si cela est rarement gênant, il faut néanmoins savoir que cette routine est grosse consommatrice de temps, et qu'il est nécessaire d'en tenir compte dans certains cas particuliers.

Un exemple : la commande SHIFT + CURSEUR HAUT provoque l'affichage de la première page de l'écran de travail et renvoie le curseur en colonne 1, ligne 1. Si la routine &BB5A était utilisée dans ce cas précis (et en général chaque fois qu'un affichage important doit se produire), le temps de remplissage de l'écran serait relativement long (entre 3 et 5 secondes en moyenne). C'est là le genre de détail qui n'a l'air de rien de prime abord, mais qui peut rendre un programme fort pénible à l'usage. Nous avons donc choisi une technique certes un peu plus complexe, mais qui présente l'avantage d'être beaucoup plus rapide (le remplissage de l'écran se fait en moins d'une seconde).

Cette technique va nous amener à explorer une zone de la mémoire ROM bien particulière que l'on appelle la *table du générateur de caractères*. Dans cette table, qui comporte 256*8 octets, sont rangées les valeurs qu'il faut écrire en mémoire écran pour représenter, en mode 2, chacun des 256 caractères existants (rappelons que les codes de contrôle ont une représentation graphique).

Toutes ces valeurs sont rangées par blocs de 8 octets chacun (en mode 2, un caractère "prend" 8 octets de la mémoire écran). La table du générateur commence à l'adresse 14336

(&3800) par les 8 octets de la représentation graphique du code de contrôle 0, viennent ensuite les 8 octets du code de contrôle 1, et ainsi de suite jusqu'au dernier caractère dont le code est 255.

On peut donc en tirer cette conclusion que l'adresse dans la table du premier des 8 octets d'un caractère peut être calculée à partir de son code ASCII et grâce à la formule suivante : adresse = 14336+(8* code ASCII).

Un exemple : le code ASCII de la lettre A est &41 (65 en décimal) ; dans la table du générateur de caractères, l'adresse du premier octet concernant cette lettre est 14336+(8*65)= 14856 (soit &3A08). Une lecture de cette adresse ROM et des 7 suivantes donne les résultats ci-après (attention, n'essayez pas de vérifier ces valeurs avec la commande PEEK, cette dernière ne permettant pas de lire en mémoire ROM) :

(&3A08) = &18

(&3A09) = &3C

(&3A0A) = &66

(&3A0B) = &66

(&3A0C) = &7E

(&3A0D) = &66

(&3A0E) = &66

(&3A0F) = &00

Ceux à qui il resterait quelques doutes pourront vérifier qu'en écrivant ces 8 valeurs dans 8 adresses de la mémoire écran situées les unes en dessous des autres, ils obtiendront bien la lettre A.

Sachant que la routine système &BC1A permet d'obtenir, à partir d'une coordonnée du mode texte, l'adresse écran correspondante, il suffira donc, pour écrire un caractère, de lire les 8 octets correspondants dans la table du générateur et de les écrire directement en mémoire écran.

La lecture de la ROM se fait grâce à la routine système &B906 (décrite précédemment) : l'adresse dans la table du générateur correspondant à un caractère spécifié ayant été calculée, la ROM inférieure est sélectionnée et la valeur contenue dans

cette adresse chargée dans le registre A ; la RAM est ensuite resélectionnée (routine &B909) et la valeur lue écrite en mémoire écran (le processus se renouvelle bien sûr 8 fois par caractère). Tout cela se passe aux lignes 740 à 749 du listing assembleur.

Dans ce contexte, les caractères redéfinis (accentués) doivent naturellement être traités d'une manière particulière. Les instructions SYMBOL AFTER et SYMBOL ont en quelque sorte pour effet de créer une seconde table dont la longueur varie en fonction du nombre de caractères redéfinis et dont l'adresse de départ dépend de la machine utilisée.

Ce programme a été écrit sur un CPC 464 équipé d'un lecteur de disquette, et l'instruction SYMBOL AFTER 236 permet de redéfinir 20 caractères (en réalité, seuls les caractères 240 à 255 le sont ; les autres sont à votre disposition...). Dans ce cas précis, la table est rangée à partir de l'adresse 42588 et l'adresse correspondant à un code spécifié s'obtient par la formule : (code ASCII–235)*8+42580. Le principe de sortie d'un caractère redéfini sera donc le même que pour les autres caractères, mais cette nouvelle table se trouvant en RAM, il va de soi que la routine de sélection de la ROM n'aura plus à être utilisée (voir les lignes 750 à 769 du listing assembleur).

Pour ce qui concerne les 664, 6128 et 464 sans lecteur de disquette, la solution consiste simplement à écrire une copie de la table à partir de l'adresse 42588. En pratique, nous nous contenterons de recopier les valeurs des caractères effectivement redéfinis (voir les lignes 348 à 374 du listing de chargement BASIC) : les 8 valeurs du caractère 240 sont chargées dans les adresses 42620 a 42627, les 8 du caractère 241 de 42628 à 42635, etc.

L'AFFICHAGE DE LA LIGNE ET DE LA COLONNE

Ces deux affichages sont effectués par le sous-programme 36 chaque fois que le curseur change de position.

Obtenir, à partir d'un programme assembleur, l'affichage d'une valeur contenue dans un registre est en principe une opération relativement complexe puisqu'il faut "traduire" cette valeur en sa représentation par des codes ASCII. Il existe

heureusement une autre solution consistant à " sauter " au beau milieu de la ROM BASIC pour utiliser un sous-programme ayant justement pour fonction d'afficher à la position actuelle du curseur système la valeur contenue dans HL (l'adresse de ce sous-programme est &EE79 pour le 464 ; pour les autres machines, voir la partie " Adaptation aux 664, 6128 et 464 sans lecteur de disquette").

Contrairement aux routines système, les sous-programmes de la ROM BASIC ne sont accessibles que par le biais du RESTART 24 (code assembleur &DF). Cette instruction, qui permet d'appeler une routine n'importe où en ROM ou en RAM, doit être immédiatement suivie de l'adresse d'un bloc de paramètres composé de 3 octets :

Octet 1 Octet faible de l'adresse de la routine qui doit être appelée

Octet 2 Octet fort de cette adresse.

Octet 3 Octet de configuration indiquant l'état RAM/ROM souhaité. Dans le cas qui nous occupe, la sélection de la ROM BASIC s'obtient par &FC (voir les lignes 915 à 921 du listing assembleur).

LA TABLE DES SÉQUENCES IMPRIMANTE

Pour éviter un allongement démesuré du programme assembleur, les codes de contrôle imprimante ont été rangés sous la forme d'une table, de manière que chacun d'eux puisse être appelé en indiquant simplement son numéro de rang dans cette table (voir en fin de listing assembleur). Notons que les caractères accentués y trouvent leur place, puisqu'ils sont obtenus grâce à des codes de contrôle (un ô, par exemple, est obtenu en sortant un o, puis en reculant d'un espace la tête d'impression, enfin en sortant un accent circonflexe).

Les différentes séquences sont séparées par &FF et il est facile, par conséquent, de rechercher n'importe quelle séquence en fonction de son rang, puis d'exécuter tous les codes qui la composent, jusqu'à la rencontre de &FF qui signalera que la séquence est terminée.

Il faut noter la petite particularité de la séquence n° 30, qui positionne la marge gauche (&9C87 à &9C89). L'emplacement

mémoire &9C89, censé contenir le numéro de la marge, est initialisé à 0. En fonction de la valeur de marge spécifiée, cet emplacement sera naturellement chargé par le programme avant l'appel de la séquence.

Rappelons enfin que les codes de contrôle utilisés sont ceux de l'imprimante DMP 2000 ; si vous souhaitez adapter le programme à une imprimante dont les codes diffèrent, vous devrez modifier la table en conséquence. Si, d'autre part, vous décidez de rajouter des séquences, il sera nécessaire d'effectuer quelques modifications mineures dans la partie du programme s'occupant de la sortie imprimante (lignes 1481 à 1673 du listing assembleur). En l'état actuel, par exemple, le sous-programme 66 considère comme illégal tout numéro de séquence supérieur à 29.

LES CODIFICATIONS PARTICULIÈRES

Les caractères accentués

Ils sont obtenus grâce aux touches du pavé numérique. Les codes correspondant à ces touches sont compris entre 128 et 138 ; lors de la saisie clavier, il leur est donc ajouté 116 pour obtenir des codes compris entre 244 et 254 (c'est-à-dire ceux qui ont été redéfinis par la commande SYMBOL).

L'espace forcé

Obtenu par la barre d'espacement pressée conjointement à la touche Shift, il est codifié par &F3 (241) lors de la saisie clavier. Il est évident qu'il sera remplacé par &20 avant toute sortie écran.

L'espace inutile

Codifié par &FF (255).

La fin de paragraphe

(Touche SHIFT + ENTER) codifié par &F2 (242).

Les codes de contrôle

Codifiés par &F1 (241) suivi de 4 octets (voir à ce sujet le paragraphe concerné du chapitre " Mode d'emploi ".

Il importe de bien se rappeler que, lors des saisies, les valeurs indiquées seront rangées sous forme ASCII, et qu'il faudra donc les transformer en valeurs réelles pour les traiter. Par exemple, le code de contrôle posant la marge gauche en colonne 18 aura l'apparence suivante :

C en vidéo inverse + 0218

En RAM, par contre, l'aspect sera différent :

&F1 Code ASCII du C en vidéo inverse (rappelons que ce caractère est redéfini).
&30 Code ASCII du 0
&32 Code ASCII du 2
&31 Code ASCII du 1
&38 Code ASCII du 8

Chaque fois qu'un de ces codes sera considéré, il faudra donc en tirer la valeur réelle qu'il représente. Le numéro de marge, par exemple, sera calculé par la formule : (&31—&30)*10+&38—&30.

LE PROGRAMME DE CHARGEMENT DES FICHIERS

Ce programme permet la création du fichier 'ASSEMBLE' (contenant le programme assembleur), et éventuellement celle du fichier 'CARACTER' (pour ceux qui possèdent un 664, un 6128 ou un 464 sans lecteur de disquette – voir à ce sujet le premier paragraphe du chapitre traitant de l'adaptation du programme à ces machines).

Il est bien sûr relativement long à entrer, mais les risques d'erreur sont pratiquement inexistants puisque chaque ligne de DATA est vérifiée. Il est certain que vous ne le rentrerez pas sans erreur du premier coup ; il est donc préférable dans un premier temps de mettre hors service les lignes de sauvegarde des fichiers (300 et 362) en utilisant un REM, et de ne les remettre en service qu'une fois toutes les erreurs corrigées.

Il reste toutefois qu'un 0 en plus ou en moins dans les lignes ne comportant que des 0 ne sera pas dépisté. C'est pourquoi nous ne saurions trop vous conseiller de vérifier en plus quelques emplacements mémoire pris au hasard en les comparant à ceux du listing assembleur (utilisez pour cela la commande POKE).

La procédure de première utilisation est expliquée au dernier paragraphe du " Mode d'emploi ".

```
1 MEMORY 36499
2 MODE 2:INK 1,0:INK 0,13:BORDER 0
3 PRINT"CHARGEMENT EN COURS"
4 '
5 '
6 ' ** GESTION ECRAN DE TRAVAIL **
7 '
8 AD=36500
9 FOR I=1 TO 188
10 READ A$,B$:V=0
11 FOR J=1 TO LEN(A$)-1 STEP 2
12 X=VAL("&"+MID$(A$,J,2)):V=V+X
13 POKE AD,X:AD=AD+1
14 NEXT J:LOCATE 1,3:PRINT 18+I
15 IF V<>VAL("&"+B$) THEN PRINT"ERREUR
EN LIGNE";18+I:PRINT CHR$(7):END
16 NEXT I
17 '
18 '
19 DATA 21983A01805236FF230B78B120,472
20 DATA F821983A0150C011E006CDC194,615
21 DATA CDCA953E2F0600CD2DBB3E1206,4AA
22 DATA 01CD2DBB3E100602CD2DBBC900,48A
23 DATA 00000000,0
24 DATA CD1BBB30FBFE7FCABF91FE8038,81B
25 DATA 09FE8B3005C674C33692FE2038,5E2
26 DATA 05FE7FDA3692FE01CAEF93FE0D,77A
27 DATA CAD393FEF03805FEFCDA3A8FFE,8F6
28 DATA 0020053EF3C33692FE1A20053E,45C
29 DATA F1C33692FEE0CA8D91FE10CA16,830
30 DATA 93FE02CAB393FE8BC8C3CC8E00,811
31 DATA 00000000000000000000000000,0
32 DATA 000000000000,0
33 DATA FEF3CA798FFEF0CA8D90FEF2CA,A52
34 DATA 4190FEF1CADA8FFEF4CA0492FE,943
35 DATA F5CA1B91FEF6CAE891FEF7CAEB,A4C
36 DATA 90C3CC8E00000000000000000000,2AD
37 DATA 00000000000000000000000000,0
38 DATA AF32803ACD29952827CDBD95CD,661
39 DATA A1952010E0132803ACD9B9520,4AE
40 DATA 06CDD195CD2290CDB28FCDCA95,7F2
41 DATA 3A803ACDFC95C3CC8ECDBD95CD,85B
42 DATA CA95C34394,2F9
43 DATA E5CD34952005E01CD6F953E01,4EF
```

```
44 DATA CD41952186 3ACDA19528 0334E1,5C7
45 DATA C93601CD9B 9528,325
46 DATA 022B342B34 E1C9,26A
47 DATA CD8F95CA43 94CDBD95CD 9B9520,7CE
48 DATA 06CDD195CD 2290CDFC8F CDCA95,83C
49 DATA 3E02CDFC95 C3CC8E,4BB
50 DATA CD34953E50 CD4195280B CD3495,590
51 DATA 300BED538A 3A18053E50 CD6F95,4BB
52 DATA 21843ACD9B 9528013423 34C9,459
53 DATA AFED5B863A 1600215100 ED52ED,56B
54 DATA 5B883A1911 500001E0C6 CDC194,560
55 DATA C900000000,C9
56 DATA AF32803ACD 1F95CA4394 CDBD95,6DC
57 DATA CDA795280F CD7490CDCA 953A80,6F7
58 DATA 3ACDFC95C3 CC8E3E0132 803ACD,6AD
59 DATA 959520E7CD DF95CDC790 18DF,78D
60 DATA 3E01CD5095 CDA7952186 3A200A,505
61 DATA 3650CD9595 28022B352B 35C9,430
62 DATA 3A853AFE01 CA4394CDBD 953E02,5F8
63 DATA 32803ACD95 952006CDDF 95CDC7,6DE
64 DATA 90CDB790CD CA953A803A CDFC95,822
65 DATA C3CC8E,21D
66 DATA 3E50CD5095 21853A35CD 9595C8,614
67 DATA 2B35C9,129
68 DATA 2A863A2600 114F0019EB 2A883A,360
69 DATA AFED521150 000150C0CD C194C9,64B
70 DATA 000000000000000000 000000,0
71 DATA CDA195CA43 94CD3495CA 43942A,705
72 DATA 883ACDBD95 3A863A4723 042288,4F3
73 DATA 3ACD349578 2804FE5020 F13D2B,53B
74 DATA 32863A2288 3AC3798F,3A1
75 DATA CD3495CA43 94ED5B8A3A 21983A,636
76 DATA 3E0101E006 09F5CDED95 FE0128,59A
77 DATA 05F1C61618 F2ED42F1CD 559111,6C0
78 DATA E0060150C0 CDC194CDCA 953E01,684
79 DATA CDFC95C3CC 8E,47B
80 DATA E5F5EB2A8A 3AED5201 5000CD7C,68C
81 DATA 9147F18032 853A783C3C 32843A,51A
82 DATA 2C7D32863A 2A8A3A2288 3AE1C9,517
83 DATA AFED423803 3C18F9ED4A 2BC900,591
84 DATA 00000000,0
85 DATA 3A873AFE01 280A3E0132 873A21,37F
86 DATA B9911807AF 32873A21BC 91E521,57F
87 DATA 190ECD75BB E106037ECD 5ABB23,591
```

```
88 DATA 10F9C3CC8E494E53202020,470
89 DATA CD1F95CA4394CD34953E01CC7E,641
90 DATA 95215F9036C9E5CD4190E136C3,701
91 DATA 2A883A36FFCDBD95CDCA95C3CC,7FB
92 DATA 8E00,8E
93 DATA CDA795CA4394CDBD953A863A3D,700
94 DATA 3DFE00CA419047CD749010FBC3,6BC
95 DATA 4190,D1
96 DATA CD1F95CA43942184 3A36022336,492
97 DATA 0123360121983A22883A117841,2FC
98 DATA CD8A94AFCDFC953E02CDFC953E,7D4
99 DATA 20CD5ABBCDCA95C3CC8E00,64B
100 DATA F53A873AFE012004F1C34A92F1,694
101 DATA 2A883A77C3798F,32E
102 DATA 32813ACD2995CA4394EB3EFFBE,6FF
103 DATA 3A813ACA4392CDCB92FE032019,5F8
104 DATA CD61953E01C46F951B3E01CD55,546
105 DATA 943A813A7713CD8A94C3798FFE,6C7
106 DATA 0128EB3E50B820E7E5D513EBED,706
107 DATA 5B8A3A3E50CD5594D1E13E01CD,621
108 DATA 55941313064FCDFE923E50CD6F,58B
109 DATA 95E5ED5B8A3A21478ACDED95FE,7C5
110 DATA FF2006228A3ACD0793E13A813A,548
111 DATA 77CD8A94C3798F0000000000,42D
112 DATA 3A863A47ED5B8A3AE5CDED95FE,77F
113 DATA 002004E13E03C97EFEFF2005EB,59A
114 DATA E13E01C97EFEF22005EBE13E02,688
115 DATA C904233E51B820DA060118D6,426
116 DATA D53EFF121310FCD1C9,4DD
117 DATA E5C521488A065036FF,428
118 DATA 2310FBC1E1C9,399
119 DATA CD2995EB200B36FFCDBD95CDCA,78C
120 DATA 95C3CC8ECD3495CA439421E292,77E
121 DATA 3618EBCDCB92E521E2923620E1,714
122 DATA FE0228153E01CD7E95233E01CD,48B
123 DATA 6F942BCD8A94CDCA95C3CC8ECD,82F
124 DATA ED95FE002845C5233E01CD6F94,5E4
125 DATA 2B3EFF12C13E01B820DFE52A8A,5CA
126 DATA 3AE5EB01500009CDED95FE0120,5D2
127 DATA 07D1E5E5D1228A3A3E50CD6F94,6B7
128 DATA 3E50CD7E95ED5B8A3ACDFE92D1,7A8
129 DATA E1CD8A94CDCA95C3CC8E3E01B8,80C
130 DATA 28C736FFCDBD9518EE,549
```

```
131 DATA 0000000000,0
132 DATA CD3495CA4394CDA795280D215F,5F5
133 DATA 9036C9CDE891215F9036C32A88,690
134 DATA 3A36F2C31693,2CE
135 DATA CD8F95CA4394CDA795CADA8F21,7EF
136 DATA 5F9036C9CDE891,434
137 DATA 215F9036C3C3DA8F,435
138 DATA CD8F95CA4394CD3495202ACDA1,6E0
139 DATA 95281D2A883A36F2CDBD953A86,5CD
140 DATA 3A473E5090CD6F95CD41953E50,5A1
141 DATA 32863AC3798F2A883A36F2C379,60D
142 DATA 8F21A68F36C921863A3E519647,531
143 DATA 3EF2C5CD4A92C13EFF10F721A6,76A
144 DATA 8F36C3C3CC8E,3A5
145 DATA 3E07CD5ABBC3CC8E0000000000,444
146 DATA 0000000000,0
147 DATA 47AF78E5D5EBED5223E5C1E1E5,8E1
148 DATA 5F160019E5D1E1E5EDB8D1E1C9,82A
149 DATA 47AF78D5E5EBED5223E5C1E1E5,8E1
150 DATA 5F1600ED52E5D1E1,44B
151 DATA E5EDB0E1D1C9,4FD
152 DATA D5E5AFEBED5223EB3A843A6F3A,742
153 DATA 863A67252DCD1ABCE519300711,462
154 DATA E006E1E519AF012FC7ED423009,5D3
155 DATA C1E1E5CDC194E1D1,65B
156 DATA C92BEBED52EB18F0,511
157 DATA D5E57ECDD194E1D11B7BB2C823,84F
158 DATA 0318F0,10B
159 DATA FEEB30225F160021000003E0819,330
160 DATA 3D20FC7CC638675059CD06B97E,5ED
161 DATA 12237AC60857300F7CD09B9C9D6,629
162 DATA EB5F1600210000 3E08193D20FC,339
163 DATA 1154A61950597E12237AC60857,41F
164 DATA 30F7C900000000000000000000,1F0
165 DATA AF2A883A11983AED52C9AF2147,59D
166 DATA 8AED5B883AED52C9AFE52A8A3A,71E
167 DATA ED5B883AED52E1C9E5D55F1600,722
168 DATA 2A883A1922883AD1E1C9E5D55F,67D
169 DATA 1600AF2A883AED5222883AD1E1,586
170 DATA C9AFE5C52A8A3A01478AED42C1,6D2
171 DATA E1C9E5D55F16002A8A3A19228A,58C
172 DATA 3AD1E1C9E5D55F1600AF2A8A3A,681
173 DATA ED52228A3AD1E1C93A853AFEFF,796
174 DATA C93A843AFE02C93A843AFE17C9,660
```

```
175 DATA 3A863AFE50C93A863AFE01C9E5,6B8
176 DATA F53A843A6F3A863A67CD75BBF1,6AB
177 DATA E1C9E52A883A7ECDAD95CD5ABB,7EA
178 DATA E1C9CDAD95CD8ABBC906012101,6BD
179 DATA 0011164F3E00CD50BCC9060021,37D
180 DATA 010011164F3E00CD50BCC9AFE5,4EB
181 DATA ED52E1C838033E01C9AF3EFFC9,6E0
182 DATA FE002814F5211904CD75BB3A85,529
183 DATA 3A6F2600CD2496F1FE01C02119,540
184 DATA 0ACD75BB3A863A6F2600CD2496,51D
185 DATA C9DF2D963E20CD5ABBC979EEFC,7D7
186 DATA 00000000000000000000,0
187 DATA CD3495CAAC96D52A8A3AAFED52,753
188 DATA 23E511973A19228A3AC1E31198,536
189 DATA 3AEDB0CD8096E1CD709621983A,761
190 DATA 117841CD8A943E01CDFC95CDCA,6E9
191 DATA 95C9,15E
192 DATA E536FF233E8ABC20F83E48BD20,63C
193 DATA F3E1C9E521843A3602233601 23,516
194 DATA 360121983A22883AE1C9,3B8
195 DATA CD3495C8EB228A3ACD7096EB01,6EE
196 DATA E00619EBCD8A94CDCA95C92198,783
197 DATA 3A228A3ACD80963E01,342
198 DATA CDFC95C3948E,443
199 DATA 3A843A473A863A2A883A3D2803,38D
200 DATA 2B18FA3E02B8200422823A,337
201 DATA C9053E5118EC,261
202 DATA 3A843AF53A863AF53E0132863A,50D
203 DATA 3E0232843A2A823AE511E00619,40B
204 DATA EBE1CD8A94F132863A,59A
205 DATA F132843ACDCA95C9,4D6
206 DATA 21983ACD7096C9CD8096C35E96,729
207 '
208 '
209 '*** SORTIE ECRAN ***
210 '
211 AD=38750
212 FOR i=1 TO 66
213 READ A$,B$:V=0
214 FOR J=1 TO LEN(A$)-1 STEP 2
215 X=VAL("&"+MID$(A$,J,2)):V=V+X
216 POKE AD,X:AD=AD+1
217 NEXT J:LOCATE 1,3:PRINT 221+I
218 IF V<>VAL("&"+B$) THEN PRINT"ERREUR
```

```
 EN LIGNE";221+I:PRINT CHR$(7):END
219 NEXT i
220 '
221 '
222 DATA CD1BBB38FBC9,39F
223 DATA CD84973A4197473A4297B82006,532
224 DATA CDA897C3D997C5CDE197C1FE01,909
225 DATA C8FE04C818E7,391
226 DATA 21599836C921D99736C9218498,5DE
227 DATA AF77237723772133977237723,47C
228 DATA 7721609A7723772377C9215998,518
229 DATA 36F521D99736CD21849836CD23,622
230 DATA 365A2336BB21339A36CD23365A,448
231 DATA 2336BB21609A36CD23365A2336,43E
232 DATA BBC900000000000,184
233 DATA CD1BBB30FBFE8BC83A8E3AFE00,71F
234 DATA 280947CD4A9821E6973618CD3D,51D
235 DATA 98CD2F982A4597E5CDD2982247,6B7
236 DATA 97E1FE0620063E01324D97C9FE,5BE
237 DATA 01C8FE022806CDDF99FE04C806,60C
238 DATA 01CD4A98CD9C98CDAD982A4797,6CB
239 DATA 22459718AD000000,1C3
240 DATA E53A8C3A673A49976FCD75BBE1,6B3
241 DATA C93A8C3A473A8D3A903C324A97,4F0
242 DATA C9E5F5C5CD5998CD7298C110F6,8C4
243 DATA F1E1C9F53A49973CFE18200BE5,70C
244 DATA C5D5CDD195D1C1E13E17324997,7A7
245 DATA F1C9,1BA
246 DATA 21449735C03A49976F2602CD75,4E4
247 DATA BB064E3E2DCD5ABB10FBCD5998,625
248 DATA 3A903A32449721E69736282142,470
249 DATA 9734C9F53A8F3AFE022802F1C9,670
250 DATA 0601CD4A98F1C93A4397FE00C8,64A
251 DATA 473A4097FE012805CD4A981803,44E
252 DATA CD3F9BAF324397C900000000000,42B
253 DATA 0000000000,0
254 DATA 0100003A4A9757CDD099FEFF28,5CE
255 DATA 0AAFB820033E01C93E04C93E20,405
256 DATA BE20032318E73EFFBE20032318,45C
257 DATA DF3EF1BE2005CD399918D53EF2,6AD
258 DATA BE200B23AFB820033E02C93E03,3E0
259 DATA C9E5C5D5CD9299FE012005D1C1,7F6
260 DATA E1180CAFBA2805F1F1F118ADF1,724
261 DATA F1F1AFB828033E05C93E06C9,58D
```

```
262 DATA E5D5C5F5233E30BE204523237E,5EC
263 DATA D63006094F8110FD47237ED630,4E0
264 DATA 804F2B2B7EFE312825FE322811,488
265 DATA 3A8C3A47047990FE0F381D7932,461
266 DATA 8D3A1817413A8D3A3C90FE0F38,449
267 DATA 0D79328C3A18073A4397813243,3A7
268 DATA 97F1C1D1E12323232323C9,573
269 DATA 04151814230C157EFEFF2818FE,442
270 DATA 202814FEF12810FEF22807AFBA,60B
271 DATA 20E83E01C9AFBA200123AFC9E5,61A
272 DATA 3A4A97916F26004806000DCD7C,3E5
273 DATA 91324B977D324C97E1C9D5ED5B,6FE
274 DATA 8A3ACDED95D1C90000000000,4AD
275 DATA F578FE012003F1180EF1FE0520,5BA
276 DATA 09F5CDB899CD0B9AF1C9F57980,836
277 DATA 3D324A973E01324B97AF324C97,467
278 DATA CD0B9AF1C93A4A97477EFEFF20,729
279 DATA 032318F8FE2020032318F1FEF1,592
280 DATA 2005CD689A18E8CD2D9AC818E2,64A
281 DATA FEF320023E20CD5ABB05C8237E,5C1
282 DATA FEF12808FEFF2804FE2020E73A,6A7
283 DATA 4B97573E20CD5ABB051520F93A,4E6
284 DATA 4C97FE00280A3D324C973E20CD,490
285 DATA 5ABB053E013CC9E5D5C53A4097,5EE
286 DATA FE012007237EFE31CCDF9BC1D1,6CE
287 DATA E12323232323C9,259
288 '
289 '
290 '*** SORTIE IMPRIMANTE ***
291 AD=39570
292 FOR I=1 TO 43
293 READ A$,B$:V=0
294 FOR J=1 TO LEN(A$)-1 STEP 2
295 X=VAL("&"+MID$(A$,J,2)):V=V+X
296 POKE AD,X:AD=AD+1
297 NEXT J:LOCATE 1,3:PRINT 302+I
298 IF V<>VAL("&"+B$) THEN PRINT"ERREUR
 EN LIGNE";302+I:PRINT CHR$(7):END
299 NEXT I
300 SAVE"ASSEMBLE",B,36500,3575
301 '
302 '
303 DATA CDB29A3A419747 3A4297B82006,563
304 DATA CDC59AC3DA9AC5CDE09AC1FE01,92F
```

```
305 DATA C8FE04C818E7,391
306 DATA 3EC921DA9A77218A9B7721689B,5F4
307 DATA 7721309B77C921DA9A36CD218A,5E6
308 DATA 9B36E521689B36E5,3F5
309 DATA 21309B36E5C9,2D0
310 DATA CD1BBBFE8BC83A8E3AFE002809,625
311 DATA 47CD3F9B21E59A3618CD3D98CD,64B
312 DATA 309B2A4597E5CDD298224797E1,6CE
313 DATA FE0620063E01324D97C9FE01C8,50F
314 DATA FE022806CDDF99FE04C8E50601,629
315 DATA CD3F9BCD749BCDAD98E1CDEE9B,8CC
316 DATA 2A4797224597 18AA,2C8
317 DATA E57832899C061ECDC49BE1C900,6AE
318 DATA 0000E5F5C5CD4E9BCD689BC110,6F6
319 DATA F6F1E1C921449735C03A903A32,6B8
320 DATA 449721E59A36282142973 43E0C,451
321 DATA CD8A9BC9,2BB
322 DATA E5F5C50601CDC49BC1F1E1C9F5,923
323 DATA 3A8F3AFE022802F1C90601CD3F,4FA
324 DATA 9BF1C90000000000,255
325 DATA E5C5FEF43809D6F247CDC49BC1,8D9
326 DATA E1C9060DFE4028F404FE5B280F,4CB
327 DATA 04FE5D280A04FE7B280504FE7D,4BA
328 DATA 2006CDC49BC1E1C9CD2EBD38FB,7A8
329 DATA CD31BDC1E1C9,426
330 DATA 210E9C3EFF23BE20FC0520F923,546
331 DATA 7EFEFFC8CD2EBD38FBCD31BD18,801
332 DATA F1237ED61EFE1ED0FE12D847CD,76E
333 DATA C49BC9ED5B4797CDED95FEFFC0,95A
334 DATA 7EFEF128032318F2CD689A18ED,699
335 DATA 00000000000000000000,0
336 DATA FF0A0DFF6F085EFF61085EFF69,618
337 DATA 087EFF75085EFF40FF5CFF7CFF,774
338 DATA 7BFF7DFF65085EFF69085EFF1B,6A9
339 DATA 5200401B5201FF1B52005B1B52,334
340 DATA 01FF1B52005D1B5201FF1B5200,3A4
341 DATA 7B1B5201FF1B52007D1B5201FF,43F
342 DATA 1B7801FF1B7800FF1B34FF1B35,4C3
343 DATA FF1B45FF1B46FF1B2D01FF1B2D,54E
344 DATA 00FF1B4DFF1B50FF1B5701FF1B,55D
345 DATA 5700FF1B6C00FF,2DC
346 '
347 '
```

```
348 ' LIGNES 353 A 374, UNIQUEMENT POUR
349 ' POSSESSEURS DE 464 SANS LECTEUR
350 ' DE DISQUETTE, DE 664 ET 6128
351 '
352 '
353 AD=42620
354 FOR I=1 TO 10
355 READ A$,B$:V=0
356 FOR J=1 TO LEN(a$)-1 STEP 2
357 X=VAL("&"+MID$(A$,J,2)):V=V+X
358 POKE AD,X:AD=AD+1
359 NEXT J:LOCATE 1,3:PRINT 364+I
360 IF V<>VAL("&"+B$) THEN PRINT"ERREUR
 EN LIGNE";364+I:PRINT CHR$(7):END
361 NEXT I
362  SAVE"CARACTER",B,42620,128
363 '
364 '
365 DATA 080C0EFFFF0E0C08FFC3999F9F,5DB
366 DATA 99C3FF020E3EFE3E0E02000000,3F5
367 DATA 003C24C0000000018245A66666618,25E
368 DATA 00304878C0C7CCC76006C003818,376
369 DATA 18183C00182466666666E0060,2DE
370 DATA 10780C7CCC760000003C66603E,392
371 DATA 08183008666666663E0006083C,278
372 DATA 667E603C00601C03C667E603C00,3AC
373 DATA 18243C667E603C001824001818,264
374 DATA 183C0000000000000180000,6C
```

LE PROGRAMME DE GESTION BASIC

Ce programme prépare la machine avant le chargement du fichier 'ASSEMBLE' (fixation de la mémoire, redéfinition de certains caractères et affichage de l'écran). La main est ensuite passée au programme assembleur qui s'occupe de toutes les saisies, les insertions, et les commandes nécessitant une grande vitesse d'exécution. Toutes les autres sont gérées directement par le BASIC, avec quelques appels ponctuels de routines assembleur.

Il doit être tapé avec beaucoup de soin, tout particulièrement lorsque des instructions POKE ou CALL sont concernées (en cas de non fonctionnement, vérifiez attentivement votre listing avant d'accuser votre machine... ou pire, l'auteur).

Les possesseurs de 664, 6128 ou 464 sans lecteur de disquette ne devront pas oublier d'effectuer les modifications décrites dans le chapitre consacré à ce sujet.

Le programme ne présente guère de particularité, et devrait facilement être compris par tous. Signalons quand même le rôle de la ligne 10, qui remédie à un défaut du BASIC de l'Amstrad. Il est en effet impossible d'utiliser la commande SYMBOL AFTER après que la mémoire a été fixée par MEMORY (l'ordinateur sortant alors un 'improper argument'... !). Cela peut parfois être fort gênant si l'on est amené à lancer plusieurs fois le programme. La solution consiste donc à utiliser un sémaphore (en l'occurrence l'emplacement mémoire 14901) qui sera mis à 1 lors du premier lancement et qui permettra par la suite d'éviter les instructions SYMBOL.

```
10 IF PEEK(14901)=1 THEN 190
20 POKE 14901,1:SYMBOL AFTER 236
30 SYMBOL 240,8,12,14,255,255,14,12,8
40 SYMBOL 241,255,195,153,159,159,153,19
5,255
50 SYMBOL 242,2,14,62,254,62,14,2,0
60 SYMBOL 243,0,0,0,50,76,0,0,0
70 SYMBOL 244,24,36,90,102,102,102,24
80 SYMBOL 245,48,72,120,12,124,204,118
90 SYMBOL 246,108,0,56,24,24,24,60
100 SYMBOL 247,24,36,102,102,102,102,62
110 SYMBOL 248,96,16,120,12,124,204,118
120 SYMBOL 249,0,0,60,102,96,62,8,24
130 SYMBOL 250,48,8,102,102,102,102,62
140 SYMBOL 251,6,8,60,102,126,96,60
150 SYMBOL 252,96,16,60,102,126,96,60
160 SYMBOL 253,24,36,60,102,126,96,60
170 SYMBOL 254,24,36,0,24,24,24,60
180 SYMBOL 255,0,0,0,0,0,24,0,0
190 MEMORY 14900
200 MODE 2:INK 0,13:INK 1,0:BORDER 0
210 PRINT STRING$(80,CHR$(143))
220 PLOT 0,26:DRAW 639,26:LOCATE 2,25:PR
INT"L=1   C=1"
230 PLOT 100,0:DRAW 100,26:PLOT 132,0:DR
AW 132,26
240 LOAD"ASSEMBLE",36500
250 '
260 '
270 '
280 '
290 ' ** FORMAT D'ORIGINE **
300 '
310 RESTORE 320:FOR I=14976 TO 14992:REA
D x:POKE I,x:NEXT
320 DATA 0,0,0,0,2,1,1,0,&98,&3a,&98,&3a
,2,78,1,1,62
330 '
340 '
350 CALL 36500
360 CALL 36556
370 '
380 ' ** MODE MENU **
390 '
```

```
400 LOCATE 20,25:PRINT"SORTIES/ENTREES(1
) FORMAT(2) SUPPRESSIONS(3) UTILITAIRES(
4)";
410 REP$=INKEY$:IF REP$="" THEN 410
420 GOSUB 1940:REP=VAL(REP$):IF REP<1 OR
 REP>4 THEN 360
430 ON REP GOTO 1050,640,470,1740
440 '
450 ' ** SUPPRESSIONS **
460 '
470 LOCATE 20,25:PRINT"SUPPRIMER: JUSQU'
AU CURSEUR(1) A PARTIR DU CURSEUR(2) TOU
T(3)";
480 REP$=INKEY$:IF REP$="" THEN 480
490 GOSUB 1940:REP=VAL(REP$):IF REP<1 OR
 REP>3 THEN 360
500 LOCATE 34,25:PRINT"CONFIRMATION ? (O
/N)";
510 R$=UPPER$(INKEY$):IF R$="" THEN 510
520 IF R$="N" THEN GOSUB 1940:GOTO 360
530 IF R$<>"O" THEN 500
540 GOSUB 1940:ON REP GOTO 550,590,600
550 IF PEEK(14982)=1 THEN 580
560 LOCATE 25,25:PRINT CHR$(7);"CURSEUR
NON POSITIONNE EN COLONNE 1 / TAPEZ UNE
TOUCHE";
570 CALL &BB06:GOSUB 1940:GOTO 360
580 CALL 38458:GOTO 360
590 CALL 38548:GOTO 360
600 CALL 38572:LOCATE 6,25:PRINT" ";:GOT
O 360
610 '
620 ' ** FORMAT **
630 '
640 LOCATE 28,25:PRINT"CREATION FORMAT (
1)    CHARGEMENT FORMAT (2)";
650 REP$=INKEY$:IF REP$="" THEN 650
660 GOSUB 1940:WINDOW#0,1,80,2,23:CLS:CA
LL 38589
670 REP=VAL(REP$):IF REP<1 OR REP>2 THEN
 360
680 ON REP GOTO 720,980
690 '
700 ' ** Creation format
```

```
710 '
720 LOCATE 20,8:INPUT"- MARGE GAUCHE (1
A 65).............:";MG
730 IF MG<1 OR MG>65 THEN LOCATE 20,8:PR
INT STRING$(50," ");CHR$(7);:GOTO 720
740 LOCATE 20,10:INPUT"- MARGE DROITE (1
6 A 80).............:";MD
750 IF MD<16 OR MD>80 THEN LOCATE 20,10:
PRINT STRING$(50," ");CHR$(7);:GOTO 740
760 LOCATE 20,12:INPUT"- MARGE HAUTE (0
A 10).............:";MH
770 IF MH<0 OR MH>10 THEN LOCATE 20,12:P
RINT STRING$(50," ");CHR$(7);:GOTO 760
780 LOCATE 20,14:INPUT"- INTERLIGNE (1 O
U 2).............:";ITL
790 IF ITL <>1 AND ITL <> 2 THEN LOCATE
20,14:PRINT STRING$(50," ");CHR$(7);:GOT
O 780
800 LOCATE 20,16:INPUT"- NOMBRE DE LIGNE
S PAR PAGE (15 A 62):";LP
810 IF LP <15 OR LP>62 THEN LOCATE 20,16
:PRINT STRING$(50," ");CHR$(7);:GOTO 800
820 IF MD-MG+1>=15 THEN 860
830 CLS:LOCATE 28,10:PRINT"*** MARGES IL
LEGALES ***";CHR$(7):LOCATE 14,12
840 PRINT"*** NOMBRE DE CARACTERES PAR L
IGNE INFERIEUR A 15 ***";CHR$(7);
850 LOCATE 32,14:PRINT"TAPEZ UNE TOUCHE"
:CALL &BB06:CLS:GOTO 720
860 POKE 14988,MG:POKE 14989,MD:POKE 149
90,MH:POKE 14991,ITL:POKE 14992,LP
870 CLS:LOCATE 25,10:PRINT"SAUVEGARDE DU
 FORMAT (O/N) ?"
880 REP$=INKEY$:IF REP$="" THEN 880
890 IF UPPER$(REP$)="O" THEN 910
900 IF UPPER$(REP$)="N" THEN 940 ELSE 88
0
910 CLS:LOCATE 20,10:INPUT"NOM DU FORMAT
 (MAXIMUM 8 LETTRES)";F$
920 IF F$="" OR F$=" " THEN PRINT CHR$(7
):GOTO 910
930 F$=LEFT$(F$,8):SAVE F$,B,14988,5
940 CLS:WINDOW#0,1,80,1,25:CALL 38619:GO
TO 360
```

```
950 '
960 '** Chargement format
970 '
980 LOCATE 20,10:INPUT"NOM DU FORMAT (MA
XIMUM 8 LETTRES)";F$
990 IF F$="" OR F$=" " THEN PRINT CHR$(7
):GOTO 980
1000 F$=LEFT$(F$,8):LOAD F$,14988
1010 GOTO 940
1020 '
1030 ' ** SORTIES/ENTREES **
1040 '
1050 LOCATE 20,25:PRINT"SORTIE ECRAN (1)
 IMPRIMANTE (2) SAUVEGARDE (3) CHARGEMEN
T (4)";
1060 REP$=INKEY$:IF REP$="" THEN 1060
1070 GOSUB 1940:REP=VAL(REP$):IF REP<1 O
R REP>4 THEN 360
1080 ON REP GOTO 1120,1280,1530,1650
1090 '
1100 '** Sortie ecran
1110 '
1120 GOSUB 1450:GOSUB 1940
1130 LOCATE 24,25:PRINT"AFFICHAGE A PART
IR PAGE";P;"/ CONFIRMATION ? (O/N)"
1140 R$=UPPER$(INKEY$):IF R$="" THEN 114
0
1150 IF R$="N" THEN GOSUB 1940:GOTO 360
1160 IF R$<>"O" THEN 1140
1170 POKE 38720,0:POKE 38886,&28
1180 WINDOW#0,1,80,2,23:CLS:WINDOW#0,1,8
0,1,25:CALL 38589:CALL 38756:POKE 14988,
MG:POKE 14989,MD
1190 IF PEEK(38733)<>1 THEN 1220
1200 GOSUB 1940:LOCATE 28,25:PRINT CHR$(
7);:PRINT"MOT TROP LONG / POUR SUITE: PE
TITE TOUCHE <ENTER>";
1210 IF INKEY(6)=0 THEN 1240 ELSE 1210
1220 GOSUB 1940:LOCATE 30,25:PRINT"RETOU
R ECRAN DE TRAVAIL: PETITE TOUCHE <ENTER
>";
1230 IF INKEY (6)=0 THEN 1240 ELSE 1230
1240 GOSUB 1940:CALL 38750:CALL 38619:GO
```

```
TO 360
1250 '
1260 ' ** Sortie imprimante
1270 '
1280 GOSUB 1450:GOSUB 1940
1290 LOCATE 24,25:PRINT"IMPRESSION A PAR
TIR PAGE";P;"/ CONFIRMATION ? (O/N)"
1300 R$=UPPER$(INKEY$):IF R$="" THEN 130
1310 IF R$="N" THEN GOSUB 1940:GOTO 360
1320 IF R$<>"O" THEN 1300
1330 POKE 38720,1:POKE 39653,&28
1340 POKE 39476,&8A:POKE 39477,&9B:POKE
39501,&8A:POKE 39502,&9B:POKE 39521,&8A:
POKE 39522,&9B
1350 PRINT#8,CHR$(27);"à";:PRINT#8,CHR$(
27);"C";CHR$(75);
1360 CALL 39570:PRINT#8:PRINT CHR$(7);
1370 IF PEEK(38733)<>1 THEN 1400
1380 LOCATE 28,25:PRINT CHR$(7);:PRINT"M
@T TROP LONG / POUR SUITE: PETITE TOUCHE
 <ENTER>";
1390 IF INKEY(6)=0 THEN 1410 ELSE 1390
1400 LOCATE 20,25:PRINT CHR$(7);:PRINT"I
MPRESSION TERMINEE - TAPEZ  PETITE TOUCH
E <ENTER>";:GOTO 1390
1410 GOSUB 1940:CALL 38750:POKE 39476,&5
A:POKE 39477,&BB:POKE 39501,&5A:POKE 395
02,&BB:POKE 39521,&5A:POKE 39522,&BB:POK
E 14988,mg:P
OKE 14989,md:GOTO 360
1420 '
1430 ' ** sp commun ecran/imprimante
1440 '
1450 GOSUB 1940:LOCATE 31,25:INPUT"A PAR
TIR DE QUELLE PAGE (MINIMUM: 1)";P
1460 IF P<1 OR P>255 THEN PRINT CHR$(7);
:GOTO 1450
1470 GOSUB 1940:POKE 38721,P:POKE 38722,
1:POKE 38723,0:POKE 38724,PEEK(14992):PO
KE 38725,&98:POKE 38726,&3A:POKE 38729,2
:POKE 38733,
0
1480 mg=PEEK(14988):md=PEEK(14989):RETUR
N
```

```
1490 '
1500 '
1510 '** Sauvegarde
1520 '
1530 GOSUB 1940:CALL 38589:WINDOW#0,1,80
,2,23:CLS:IF B$="" THEN 1560
1540 LOCATE 20,9:PRINT"*** DERNIER NOM E
N MEMOIRE: ";B$;" ***"
1550 LOCATE 15,14:PRINT"*** SI LE NOM ES
T IDENTIQUE, TAPEZ <ENTER> ***"
1560 LOCATE 15,12:INPUT"NOM DU TEXTE A S
AUVEGARDER (8 LETTRES MAXIMUM) ";A$
1570 IF A$=" " THEN PRINT CHR$(7);:GOTO
1560
1580 IF A$="" AND B$="" THEN PRINT CHR$(
7);:GOTO 1560
1590 IF A$="" THEN A$=B$ ELSE B$=LEFT$(A
$,8)
1600 FIN=256*PEEK(14987)+PEEK(14986):LON
G=FIN-14985:SAVE A$,B,14986,LONG
1610 WINDOW#0,1,80,1,25:CALL 38619:GOTO
360
1620 '
1630 '** Chargement
1640 '
1650 GOSUB 1940:CALL 38589:WINDOW#0,1,80
,2,23:CLS
1660 LOCATE 15,10:INPUT"NOM DU TEXTE (8
LETTRES MAXIMUM)";B$
1670 IF B$="" OR B$=" " THEN CLS:PRINT C
HR$(7);:GOTO 1660
1680 IF LEN(B$)>8 THEN CLS:PRINT CHR$(7)
;:GOTO 1660
1690 B$=LEFT$(B$,8):CALL 38662:LOAD B$,1
4986
1700 WINDOW#0,1,80,1,25:CALL 38619:GOTO
360
1710 '
1720 '** UTILITAIRES **
1730 '
1740 LOCATE 26,25:PRINT"VOIR CONTENU DIS
QUETTE (1)   RETOURNER AU BASIC (2)"
1750 REP$=INKEY$:IF REP$="" THEN 1750
1760 REP=VAL(REP$):IF REP<1 OR REP>2 THE
```

```
N GOSUB 1940:GOTO 360
1770 ON REF GOTO 1810,1870
1780 '
1790 ' ** Contenu disquette
1800 '
1810 WINDOW#0,1,80,2,23:CLS:CALL 38589:C
AT
1820 WINDOW#0,1,80,1,25:GOSUB 1940:LOCAT
E 30,25:PRINT"RETOUR ECRAN DE TRAVAIL: T
APEZ UNE TOUCHE";
1830 CALL &BB06:CALL 38619:GOSUB 1940:GO
TO 360
1840 '
1850 ' ** Retour basic
1860 '
1870 GOSUB 1940:LOCATE 34,25:PRINT"CONFI
RMATION ? (O/N)";
1880 R$=UPPER$(INKEY$):IF R$="" THEN 188
0
1890 IF R$="N" THEN GOSUB 1940:GOTO 360
1900 IF R$<>"O" THEN 1880
1910 CALL 0
1920 '
1930 '
1940 LOCATE 20,25:PRINT STRING$(61,CHR$(
32));:RETURN
1950 '
1960 '
1970 ' *** RATTRAPAGE DU TEXTE EN CAS D'
ERREUR ***
1980 '
1990 INPUT"NOM DU TEXTE A SAUVEGARDER (8
 LETTRES MAXIMUM) ";A$
2000 FIN=256*PEEK(14987)+PEEK(14986):LON
G=FIN-14985:SAVE A$,B,14986,LONG
```

ADAPTATION AUX 664, 6128 ET 464 SANS LECTEUR DE DISQUETTE

Ce programme, qui a été mis au point sur un 464 équipé d'un lecteur de disquette, est adaptable sans aucune difficulté sur les autres CPC.

■ MODIFICATION COMMUNE AUX TROIS MACHINES

Cette modification commune concerne naturellement l'affichage des caractères redéfinis, ainsi qu'il a été expliqué dans les commentaires du chapitre " Programme assembleur " (s'y reporter éventuellement).

Les possesseurs de l'une de ces machines devront tout d'abord entrer l'intégralité du programme de chargement BASIC, y compris, donc, les lignes 353 à 374 : le fichier créé par ces lignes représente une copie exacte de la table de caractères redéfinis telle qu'elle est mise en place par l'instruction SYMBOL AFTER.

Ce fichier devra naturellement être chargé à partir de l'adresse 42620, dès le début du programme BASIC de gestion du traitement de texte ; cela se fait très facilement en ajoutant la ligne suivante :

260 LOAD " CARACTER ", 42620

■ MODIFICATION SPÉCIFIQUE AUX 664 ET 6128

Sur ces deux machines, la ROM BASIC est presque identique à celle du 464, mais les adresses sont quelque peu décalées. Il sera donc nécessaire de modifier l'adresse de la routine BASIC utilisée dans le sous-programme 36 qui réalise l'affichage de la ligne et/ou de la colonne (voir le " Programme assembleur " et les commentaires à ce sujet).

Cette adresse étant rangée dans les emplacements mémoire 38445 (&962D) et 38446 (&962E) du bloc assembleur (voir ligne 920 du listing), il est tout à fait possible de la modifier après le chargement de la ligne 240 du programme de gestion BASIC, en ajoutant une ligne 250.

Pour le 664 :

250 POKE 38445,73 : POKE 38446,239

Pour le 6128 :

250 POKE 38445,68 : POKE 38446,239

■ MODIFICATIONS SPÉCIFIQUES AU 464 SANS LECTEUR DE DISQUETTE

Il est bien évident qu'un lecteur de cassette est infiniment moins agréable et moins souple d'utilisation qu'un lecteur de disquette, pour ce qui concerne tant la rapidité que la sécurité, surtout avec des programmes de ce genre où les sauvegardes et chargements sont en principe fréquents (c'est d'ailleurs probablement la raison pour laquelle on ne trouve pas sur le marché — à l'heure où ces lignes sont écrites, en tout cas — de traitement de texte sur cassette pour Amstrad.

Que les adeptes des cassettes se rassurent, pourtant, puisque les durées de chargement restent tout à fait raisonnables : environ une minute et demie pour le programme de gestion BASIC, 50 secondes pour le fichier 'ASSEMBLE', et une dizaine de secondes pour le fichier 'CARACTER'. Le temps nécessaire pour être opérationnel est donc loin d'être rédhibitoire.

D'autre part, et à titre indicatif, il faudra compter entre 4 et 5 minutes pour sauvegarder ou charger un texte occupant toute la RAM bibliothèque (soit 20 400 caractères y compris les espaces inutiles, ce qui représente déjà un texte conséquent). Assurez-vous qu'il reste suffisamment de bande avant toute sauvegarde, et en cas de doute n'hésitez pas à changer de cassette.

Rappelons également qu'il est toujours possible de doubler les vitesses d'écriture (et donc de lecture) sur cassette grâce à l'instruction SPEED WRITE (reportez-vous éventuellement au guide de l'utilisateur).

Reste naturellement le problème de la sécurité : rares sont ceux qui n'ont pas un jour ou l'autre appuyé sur REC+PLAY au

lieu de PLAY, ou à qui il n'est pas arrivé d'enregistrer un fichier sur un autre. Que vous dire, sinon d'éviter le désordre dans vos cassettes (c'est difficile, c'est vrai...), de bien noter vos compteurs et d'être extrêmement attentif lors de toute opération impliquant la cassette. Si malgré tous ces bons conseils, les choses tournaient mal, rappelez-vous qu'il est toujours possible de sauver un texte en utilisant la procédure de secours décrite à l'avant-dernier paragraphe du " Mode d'emploi ".

En plus de la modification signalée au début du présent chapitre, il sera nécessaire d'ajouter trois lignes au programme de gestion BASIC, et d'en modifier quelques autres. En réalité, le programme pourrait fonctionner tel quel, mais quelques affichages et coups de cloche supplémentaires seront certainement les bienvenus pour éviter toute étourderie.

Lignes à ajouter :

```
235 PRINT CHR$(7);:LOCATE 1,3:PRINT " PREPAREZ CASSETTE
POUR FICHIER ' ASSEMBLE ' "

255 PRINT CHR$(7);:PRINT " PREPAREZ CASSETTE POUR FICHIER
' CARACTER ' "

1595 LOCATE 1,16
```

Il est également souhaitable d'insérer un coup de cloche chaque fois que le lecteur de disquette doit être utilisé, et avant l'affichage par le système des messages ' PRESS PLAY THEN ANY KEY ' et ' PRESS REC+PLAY THEN ANY KEY ' :

```
930 F$=LEFT$(F$,8):PRINT CHR$(7);:SAVE F$,B,14988,5

1000 F$=LEFT$(F$,8):PRINT CHR$(7);:LOAD F$,14988

1600 FIN=256*PEEK(14987)+PEEK(14986):LONG=FIN-14985:PRINT
CHR$(7);:SAVE A$,B,14986,LONG

1690 B$=LEFT$(B$,8):CALL 38662:PRINT CHR$(7);:LOAD B$,14986
```

Il faudra enfin remplacer le mot ' DISQUETTE ' par ' CASSETTE ' dans la ligne 1740. A ce sujet, signalons qu'il est possible d'arrêter la lecture de la cassette en procédant comme

suit : attendre une pose entre deux blocs de données (il est très facile de s'en rendre compte au bruit), tapez une fois la touche BREAK, puis une fois la petite touche ENTER. Toute autre procédure pourrait entraîner des complications.

La procédure de première utilisation reste la même (voir mode d'emploi), sauf qu'en plus du fichier ' ASSEMBLE ' il sera également nécessaire de charger le fichier ' CARACTER '.

Pour simplifier la mise en route du traitement de texte, l'ordre de rangement sur la cassette sera naturellement :

1. Le programme BASIC de gestion.
2. Le fichier binaire ' ASSEMBLE '.
3. Le fichier binaire ' CARACTER '.

Le programme BASIC une fois chargé, la cassette se retrouvera positionnée au début des fichiers binaires et vous n'aurez plus à manipuler quoi que ce soit.

Il est bien sûr fortement conseillé d'utiliser une autre cassette pour enregistrer vos textes.

POUR UN CATALOGUE COMPLET
DE NOS PUBLICATIONS

FRANCE
6-8, Impasse du Curé
75881 PARIS CEDEX 18
Tél. : (1) 42 03 95 95
Télex : 211801

U.S.A.
2344 Sixth Street
Berkeley, CA 94710
Tel. : (415) 848.8333
Telex : 336311

ALLEMAGNE
Vogelsanger. WEG 111
4000 Düsseldorf 30
Postfach N° 30.09.61
Tel. : (0211) 61 80 2-0
Telex : 08588163

Paris • Berkeley • Düsseldorf

Achevé d'imprimer par l'imprimerie PRAXIE
Dépôt légal 3e trimestre 1986 - Imprimeur n° 998

Le traitement de texte présenté dans cet ouvrage, pour l'essentiel en assembleur, est destiné à tous les possesseurs de CPC 464 (avec ou sans lecteur de disquette), 664 et 6128. Mis au point avec une imprimante DMP 2000, il peut facilement être adapté à d'autres imprimantes.
Il dispose de toutes les commandes importantes de gestion de l'écran de travail (insertion, suppression de lignes ou de blocs, renvoi du curseur en début ou en fin de texte, définition des formats, etc.), permet tous les caractères accentués du français et, naturellement, la justification en sortie écran ou imprimante.
Des codes de contrôle permettent l'accès aux différentes options de l'imprimante (caractères gras, italiques, soulignés, NLQ, etc.), ainsi que des modifications de format à l'intérieur du texte.
La RAM bibliothèque a une capacité de 20 400 caractères, ce qui peut représenter entre 5 et 15 pages selon le format et la présentation choisis (le nombre de colonnes peut varier de 15 à 80).
Tous ceux qui s'intéressent à l'assembleur pourront facilement étudier le listing abondamment commenté et, pourquoi pas, rendre le programme encore plus performant en y ajoutant des commandes de leur propre cru.

0221 0986 128 F
9 782736 102210

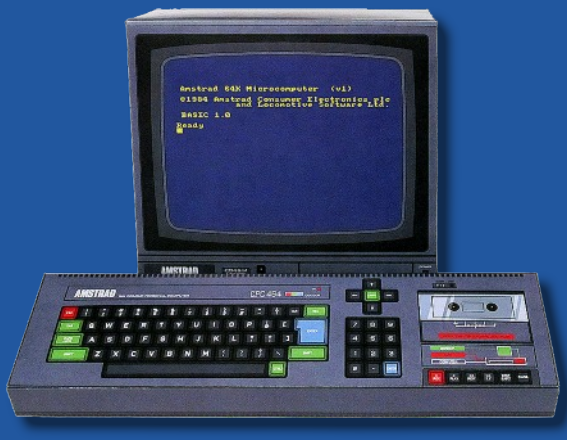
SYBEX

AMSTRAD PROGRAMMEZ VOTRE TRAITEMENT DE TEXTE

Document numérisé avec amour par :

AMSTRAD CPC

MÉMOIRE ÉCRITE



ACME

https://acpc.me/