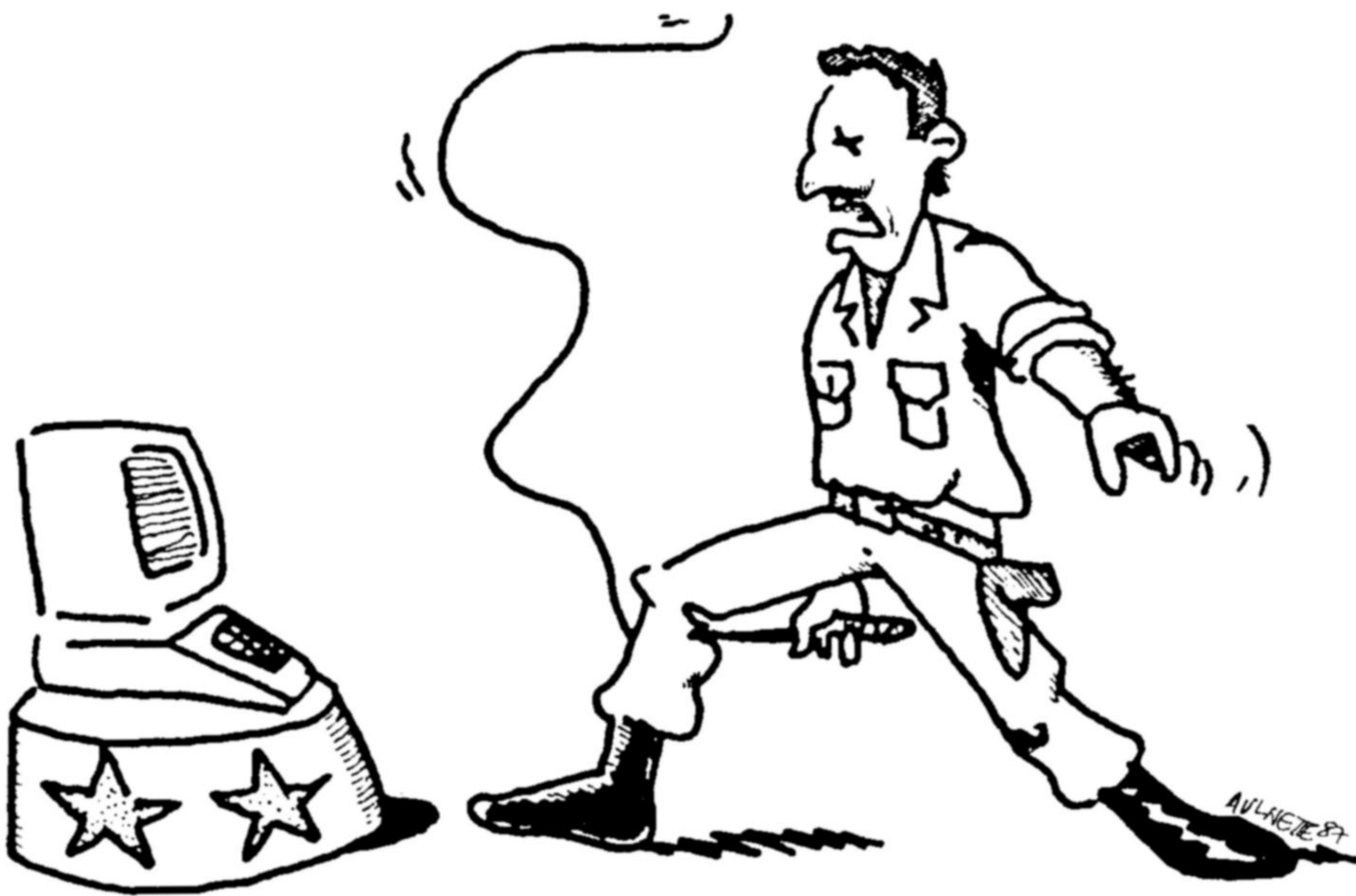


# EDIT 2

Bernard BESSE

Lorsqu'un programme BASIC en cours de développement atteint une certaine dimension, les possibilités de l'éditeur intégré deviennent insuffisantes et le besoin d'un éditeur de texte commence à se faire sentir.



Mais ce que l'on gagne en puissance avec un éditeur séparé, on le perd en souplesse car, même pour des opérations limitées, il faut :

- sauvegarder en ASCII (s'il ne l'est déjà) le programme à "éditer",
- charger l'éditeur,
- charger le programme à éditer.

Et à la fin du travail :

- sauvegarder le programme édité,
- charger le programme édité.

Contrairement aux éditeurs séparés, l'éditeur intégré opère en "temps réel" puisque, sitôt une modification effectuée, il est possible d'exécuter le programme, sans avoir à charger ou sauvegarder quoi que ce soit.

Le programme EDIT2 a pour but d'ajouter aux possibilités de l'éditeur intégré les fonctions temps réel suivantes :

- trouver toutes les occurrences d'un NOM de variable,
- trouver toutes les occurrences d'un NUMERO de ligne.

Ces fonctions permettent notamment :

- de créer un nom de variable en toute connaissance de cause,
- de vérifier si les noms des variables figurant dans une routine à MERGE risquent d'interférer avec ceux déjà présents dans le programme,
- de changer d'une façon sûre le nom d'une variable,
- de s'assurer qu'une ligne n'est pas référencée avant de la supprimer.

## MISE EN OEUVRE

EDIT2 est stocké en format ASCII à partir de 60000.

Après chargement du programme en cours de développement :

- créer une ligne 1 GOTO 60000
- faire MERGE "edit2"

Faire RUN (ou RUN 60000 s'il n'y a pas la ligne 1 déjà citée), puis :

- choisir la sortie sur imprimante ou sur écran,

- entrer un nom de VARIABLE ou un NUMERO de ligne (pour revenir à la question précédente "Imprimante ou Ecran", faire seulement ENTER).

La variable peut être écrite en majuscules ou en minuscules (ou un mélange des deux).

On ajoute les suffixes %, !, \$ s'il y a lieu et, si la variable est indicée, une parenthèse gauche.

Exemple : bldULE\$ [sans espaces]

Renumérotation du programme édité :

- on peut, au cours du traitement, faire RENUM sans être obligé de rétablir la numérotation d'EDIT2 à partir de 60000

D'où l'intérêt de la ligne initiale 1 GOTO 60000

EDIT2 s'arrête par BREAK.

## POINTS PARTICULIERS

La recherche se fait principalement par INSTR et non par PEEK (c'est beaucoup plus rapide).

Le procédé utilisé est celui de la chaîne forcée, on force dans les 3 octets du descripteur d'une variable ZZ\$ (descripteur créé en faisant ZZ\$ = "") :

- la longueur de la ligne,
- l'adresse de la ligne (2 octets).

Le numéro d'une ligne référencée peut exister sous deux formes :

- le numéro lui-même en binaire sur 2 octets,
- l'adresse de la ligne en binaire sur 2 octets.

La routine ADRLINE a pour but de trouver rapidement l'adresse de la ligne dont on recherche les occurrences.

EDIT2 s'arrête, non pas sur le test NUMERO ligne  $\geq 60000$ , mais sur le test ADRESSE ligne  $\geq$  ADRESSE 1ère d'EDIT2, ce qui permet l'utilisation de RENUM.

A la fin d'une séance, si la mise au point n'est pas terminée, il est souvent souhaitable de SAUVEGARDER ensemble le PROGRAMME + EDIT2. On est ainsi prêt à opérer dans les meilleurs délais à la séance suivante.

Pour traiter EDIT2 par EDIT2 :

- charger EDIT2,
- RENUM 62000,
- MERGE "EDIT2",
- créer ligne : 1 GOTO 62000,
- RUN.

L'attention de nos fidèles lecteurs est attirée sur les routines machine KDEEK et KDOKE (1<sup>er</sup> K pour déKalage) plus versatiles que DEEK et DOKE.

```

60000 :GOTO 60050:THEN 61130
*** Ne pas enlever cette lere Ligne ***
60010 p$="edit2":PRINT CHR$(7)CHR$(PEEK(
&A700)+65) : "p$ Ascii":SAVE p$,a:END
60020 '-----
60030 '(c) Bernard BESSE
60040 '-----
60050 :
60060 CLEAR:ON BREAK GOSUB 61110:MODE 2:
DEFINT a-z:vr=-1:fa=0:zz$=""
60070 ' CALL @kdeek(1), Adresse de Base,
  Deplacement, @R1 (R1=Val. DEEK)
60080 DATA DD,6E,04,DD,66,05,DD,5E,02,DD
,56,03,19,5E,23,56
60090 DATA DD,6E,00,DD,66,01,73,23,72,C9
60100 RESTORE 60080:DIM kdeek(13):r1=0
60110 FOR k=1 TO 13:READ l$,m$:kdeek(k)=
VAL("&H"+m$+l$):NEXT k
60120 ' CALL @kdoke(1), Adresse de Base,
  Deplacement, Valeur a DOKER
60130 DATA DD,6E,04,DD,66,05,DD,5E,02,DD
,56,03,DD,4E,00,DD
60140 DATA 46,01,19,71,23,70,C9,00
60150 RESTORE 60130:DIM kdoke(12)
60160 FOR k=1 TO 12:READ l$,m$:kdoke(k)=
VAL("&H"+m$+l$):NEXT k
60170 ' CALL @adrline(1), NUM, @R1 (R1=
adr.de la ligne NUM)
60180 DATA DD,4E,02,DD,46,03,FD,21,6F,01
,FD,6E,03,FD,66,04
60190 DATA B7,ED,42,28,14,30,0E,FD,5E,01
,FD,56,02,7B,82,28
60200 DATA 04,FD,19,18,E5,FD,21,00,00,FD
,E5,D1,DD,6E,00,DD
60210 DATA 66,01,73,23,72,C9
60220 RESTORE 60180:DIM adrline(27):r1=0
60230 FOR k=1 TO 27:READ l$,m$:adrline(k
)=VAL("&"+m$+l$):NEXT k
60240 RESTORE 60000:CALL@kdeek(1),&AE30,
0,@r1:adr60000=r1' adr lere ligne
60250 :
60260 CLS:PRINT"<I>mprimante ou <E>cran
? ";
60270 WHILE INKEY$<>"" :WEND
60280 ie$=UPPER$(INKEY$):IF ie$="" THEN 6
0280
60290 IF ie$="E" THEN ca=0 ELSE IF ie$="I
" THEN ca=8 ELSE PRINT CHR$(7):GOTO 60250
60300 PRINT ie$
60310 :
60320 deb=368:num=fa:ref=0

```

```

60330 INPUT"Nom (VARIABLE) ou Numero (LI
GNE) ";name$:nm$=name$
60340 IF name$="" THEN 60250
60350 IF INSTR(name$," ")>0 THEN PRINT C
HR$(7);:GOTO 60310
60360 name$=UPPER$(name$):lv=LEN(name$):
pg=fa
60370 car.ok$=".ABCDEFGHIJKLMNPOQRSTUVWXYZ
1234567890"
60380 xx=INSTR(car.ok$,LEFT$(name$,1)):I
F xx<2 THEN PRINT CHR$(7);:GOTO 60310
60390 IF xx<28 THEN 60510
60400 IF VAL(name$)=0 OR VAL(name$)>6553
5 THEN PRINT CHR$(7);:GOTO 60310
60410 FOR k=2 TO lv
60420 IF INSTR(car.ok$,MID$(name$,k,1)
)<28 THEN PRINT CHR$(7);:GOTO 60310
60430 NEXT k
60440 zline!=VAL(name$):IF zline!>32767
THEN zline=zline!-65536 ELSE zline=zline
!
60450 lsb=PEEK(@zline):msb=PEEK(@zline+1
)
60460 look2$=CHR$(30)+CHR$(lsb)+CHR$(msb
)
60470 CALL@adrline(1),zline,@r1
60480 lsb=PEEK(@r1):msb=PEEK(@r1+1)
60490 look1$=CHR$(29)+CHR$(lsb)+CHR$(msb
)
60500 num=vr:GOTO 60680
60510 :
60520 IF RIGHT$(name$,1)="(" THEN pg=vr:n
ame$=LEFT$(name$,lv-1):lv=lv-1
60530 IF RIGHT$(name$,1)="% " THEN type=2:
GOTO 60570
60540 IF RIGHT$(name$,1)="$ " THEN type=3:
GOTO 60570
60550 IF RIGHT$(name$,1)="!" THEN type=4:
GOTO 60570
60560 type$="":GOTO 60590
60570 :
60580 type$=CHR$(type):name$=LEFT$(name$
,lv-1):lv=lv-1
60590 :
60600 IF lv=1 THEN 60640
60610 FOR k=2 TO lv
60620 IF INSTR(car.ok$,MID$(name$,k,1)
)=0 THEN PRINT CHR$(7);:GOTO 60310
60630 NEXT k
60640 :
60650 der$=RIGHT$(name$,1):name$=LEFT$(n
ame$,lv-1):lv=lv-1
60660 der=ASC(der$):maju=der+128:IF der>
=65 AND der<=90 THEN minu=maju+32 ELSE m
inu=maju
60670 look2$=type$+STRING$(2,0)
60680 :
60690 '--- Recherche et Affichage ---
60700 PRINT#ca,"LIGNES contenant VARIABLE

```

```

E (ou LIGNE): ";nm$
60710 PRINT#ca,"-----
-----";
60720 lg=0
60730 :
60740 deb=deb+lg:CALL@kdeek(1),deb,0,@r1
:lg=r1:IF lg=0 THEN 61080
60750 CALL@kdeek(1),deb,2,@r1:nline=r1
60760 IF nline>0 THEN nline!=nline ELSE
nline!=65536+nline
60770 IF deb>adr60000 THEN 61080
60780 IF lg>255 THEN PRINT#ca:PRINT#ca,"
Reduire long.ligne";nline!;"SVP":END
60790 POKE@zz$,lg:CALL@kdoke(1),@zz$,1,d
eb
60800 c1=1
60810 :
60820 IF NOT num THEN 60870
60830 yy=INSTR(c1,zz$,look1$):xx=INSTR(c
1,zz$,look2$)
60840 IF(xx OR yy)=0 THEN 60730
60850 IF yy>0 THEN IF xx=0 THEN xx=yy EL
SE xx=MIN(xx,yy)
60860 GOTO 60890
60870 :
60880 xx=INSTR(c1,zz$,look2$):IF xx=0 TH
EN 60730
60890 :
60900 c1=xx+1
60910 IF num THEN 61050
60920 ad=deb+xx+lv+2:IF type$="" THEN ad=
ad-1
60930 IF PEEK(ad)<>maju AND PEEK(ad)<>mi
nu THEN 60810
60940 IF UPPER$(MID$(zz$,xx+LEN(look2$),
lv))<>name$ THEN 60810
60950 FOR k=ad+1 TO deb+lg-1
60960 IF PEEK(k)=32 THEN 60990
60970 IF PEEK(k)=40 THEN IF pg THEN 61
020 ELSE 60810
60980 IF pg THEN 60810 ELSE 61020
60990 :
61000 NEXT k
61010 IF pg THEN 60810
61020 :
61030 IF type$<>"" THEN 61050
61040 type=PEEK(deb+xx-2):IF type<11 OR
type>13 THEN 60810
61050 :
61060 IF ref MOD 13=0 THEN PRINT#ca
61070 PRINT#ca,USING"#####";nline!;:ref
=ref+1:GOTO 60810
61080 :
61090 PRINT#ca:PRINT#ca,"-----
-----"
61100 GOTO 60310
61110 :REM<BREAK>
61120 CLEAR:END:RETURN
61130 : ** LIGNE FIN - Ne pas enlever **

```