MÉMOIRE DE MASSE
BIDOUILLE DU SYSTÈME THOMSON
Suite de l'article de P. DEBART
Bulletin n° 44 pages 206 à 219

P. DEBART

VIII) PROGRAMME BASIC CASSETTE TROP LONG

Un certain nombre de programmes basic utilisent plus de 22 Ko et ne peuvent loger avec le DOS: C'est parfois une forme de protection (exemple: AUTOBAS d'Informatique Pour Tous).

Quatre possibilités :

- 1) Utiliser le minidos disquette mis au point pour les cartouches. On ne peut que faire `LOAD`, pas de `DIR` ni de `SAVE`. Charger le programme sur disquette avec la cartouche assembleur ou avec un TO9.
- 2) Utiliser le basic 128.
- 3) Souvent ces programmes contiennent beaucoup de données. Les placer dans un fichier en remplaçant les `READ DATA` par des `INPUT#1`, avec disquette l'accès à ces fichiers est facile.
- 4) Découper si possible le programme en plusieurs parties, en dupliquant les sous-programmes communs. Finir la première partie par `LOAD"PROG2",R`.

X) AMÉLIORATION DES PROGRAMMES

Écran

Toute utilisation de l'informatique doit viser transparence et confort d'utilisation.

Hélas, pour des raisons essentiellement commerciales, I.P.T. s'est orienté dans le sens contraire.

Il faut donc reprendre les programmes et en supprimer tout ce qui marche mal, volontairement ou non:

Curseur

Allumé: `LOCATE X,Y,1` ou `PRINT CHR$(17);`

éteint : `LOCATE X,Y,0` ou `PRINT CHR$(20);`

Lors des compositions graphiques ou si l'on utilise le crayon optique le curseur doit être éteint. Le laisser allumé lors des accès clavier:

Le dernier paramètre de `LOCATE` n'est modifié qu'explicitement. Il n'est pas nécessaire de le répéter. Le moniteur a quelques bugs et perd parfois le curseur, mais multiplier les `LOCATE X,Y,1` ne change rien.

Attribut

La taille des caractères, transitoire, est réinitialisée lors du OK. Il est donc inutile de commencer les programmes par `ATTRB 0,0`.

Accents

La longueur d'une lettre accentuée est 3 (Invention PTT, norme télétel). Cela induit des bugs dans les comparaisons de chaînes de caractères, le calcul de leur longueur et à l'affichage:

Par exemple il est impossible d'écrire un accent colonne 38 ou 39:

`LOCATE 38,Y: PRINT"é"` affiche é au début de la ligne suivante.

Une chaîne contenant des accents sera affichée à la ligne suivante si la position d'affichage plus la longueur dépasse 39: La frappe de la touche ACC colonne 39 sur TO7 chaîne la ligne avec la suivante (effacer ensuite la lettre, le chaînage reste). Cela remplace le CNT W du MO 5.

Ligne 24 : (25ème ligne)

Pour écrire ligne 24 utiliser `PRINT....;` le `";"` empêche le défilement de l'écran. Il est impossible d'écrire avec `PRINT` dans la case 39,24. Utiliser `PSET(39,24)"."`.

`LOCATE X,Y` écrit ligne 24 si Y est supérieur à 24 mais génère une erreur sur TO9. Attention aux `LOCATE 0,25` de certains programmes qui ont oublié la ligne 0.

De même `LOCATE 40,Y`; `PSET(X,200)` ou `PSET(320,Y)` acceptés par le TO7 génèrent une erreur en BASIC 128.

XI) AMÉLIORATION DES PROGRAMMES

Clavier

L'accès clavier doit être le plus transparent possible: Exactement le contraire des programmes Informatique Pour Tous qui commencent par `SCREEN0,0,0:LOCATE0,0,0` et où il faut taper une touche sans que cela soit précisé. C'est se moquer de nos élèves:

Pour corriger, avant la boucle `INKEY$` écrire un message et initialiser `INKEY$` avec un premier INKEY$ fictif

```
100 PRINT"Pour commencer taper sur une touche";
110 R$=INKEY$
120 R$=INKEY$:IF R$="" THEN 120
```

Ces deux dernières lignes sont avantageusement remplacées par: `R$=INPUT$(1)`, mais le CNT C n'est plus actif. Bonne occasion de le contrôler et de remettre les attributs en place:

```
R$=INPUT$(1)
IF R$=CHR$(3) THEN CONSOLE 0,24:LOCATE 0,24,1:PRINT"Au revoir":END
```

Si possible, remplacer `INPUT` par un sous programme testant les caractères un par un, permet un contrôle strict, des réponses plus rapides et en numérique évite des "Redo from start", mal compris des élèves.

Pour saisir un seul caractère, utiliser `INPUT$(1)` à la place de `INPUT`.

Tester une possibilité d'accès direct (touche RAZ) et éventuellement de guide (touche ACC, norme télétel) :

```
R$=INPUT$(1)
IF R$=CHR$<1) THEN PRINT"appeler votre professeur":STOP'CNT A
IF R$=CHR$(3) THEN CONSOLE 0,24:LOCATE 0,24,1:END'CNT C
IF R$=CHR$(12) THEN 1000'
IF R$=CHR$(22) THEN GOSUB 2000'ACC
1000 'MENU ....
2000 'GUIDE ....
```

Ne jamais utiliser des `INPUT` multiples : `INPUT A,B`.

Utiliser LINEINPUT si l'on veut saisir des phrases où peuvent figurer des virgules.

Ne pas demander à l'utilisateur de se mettre en majuscule ou minuscule mais l'imposer en utilisant par exemple POKE PRC,PEEK(PRC) AND 247 (PRC=&HE7C3 ou &HA7C0).

Éviter les consignes trop strictes de la forme "pour commencer taper sur C", remplacer par "une touche" (quelconque). Précéder toute lecture d'un message. Pour les non-initiés préciser lorsque l'on doit terminer par ENTRÉE: Dire clairement si l'accès est au crayon optique et/ou au clavier. Tester éventuellement les deux possibilités.

XII) AMÉLIORATIONS DIVERSES DES PROGRAMMES BASIC

Boucle d'attente

Attention aux longueurs. Un des problèmes de l'informatique, surtout en basic, est la lenteur. Certains programmeurs en rajoute, en abusant du graphisme, en multipliant la musique ou des : FOR I=1 TO 1000 :NEXT.

Dans les boucles d'attente, permettre à l'utilisateur, des sorties plus rapides, en testant clavier et/ou crayon optique. Attention la vitesse d'exécution varie en fonction des appareils:

```
FOR I=1 TO 8000:NEXT'10 secondes
```

est à remplacer, en utilisant PLAY (vitesse invariable), par :

```
R$=INKEY$:PLAY"L1"
FOR I=1 TO 300
PLAY"P":R$=INKEY$
IF R$>"" OR PTRIG THEN I=300
NEXT I:PLAY"L24"
```

Modifier de même les sous-programmes TIME$ de temporisation Michel Oury qui utilise le TIMER du TO7, incompatibles avec les TO9 et MO5.(1/10ème de seconde à remplacer par PLAY"L1P")

Ordinateur que fais-tu ?

Ne pas laisser tourner un programme longtemps sans affichage. Dans un calcul assez long, montrer que l'ordinateur travaille, avec environ toutes les 10 secondes, un message ou tout simplement un SCREEN,,X.

```
10 FOR I=1 TO 1000
20 IF I MOD 100=0 THEN SCREEN „ (Ià100) MOD 8
```

Signaler les appels aux périphériques (Attention à la sélection par défaut). Faire précéder :

```
200 LOADM"TOTO"
```

par le message :

```
195 PRINT"Je charge le fichier TOTO"
```

Avec ce genre de message le programme "MYCOTER" sur les devient utilisable (actuellement on n'a pas la patience d'attendre le chargement de fichiers) :

Ne pas oublier de gérer les erreurs disque `IF ERL=200 THEN 'erreur au chargement de...'`

RND

Le basic Thomson n'a pas de fonction `RANDOMIZE`. Pour rendre `RND` "aléatoire" il n'est pas nécessaire de demander un nombre mais utiliser après un affichage une boucle `INKEY$`

```
100 PRINT"taper une touche";:R$=INKEY$
110 R$=INKEY$:A=RND:IF R$="" THEN 110
```

FOR ...NEXT

En basic Microsoft il est interdit de sortir d'une boucle sans passer par l'instruction `NEXT`

```
10 FOR 1=1 TO 10
30 IF CONDITION THEN 110
100 NEXT I
110 'SUITE
```

Doit être remplacé par :

```
30 IF CONDITION THEN I=10:GOTO 100
```

Si la valeur de I est à réutiliser, la ranger dans un drapeau :

```
30 IF CONDITION THEN DRAP=I:I=10:GOTO 100
110 I=DRAP
```

En BASIC 128 on peut écrire :

```
30 IF CONDITION THEN EXIT
```

Les erreurs ne sont pas signalées immédiatement: Uniquement lorsque que l'on retrouve un NEXT sans nom de variable ou un FOR .. NEXT concernant la même variable. Les messages ne sont explicites : NF ou FN, parfois SN error.

Analyse d'erreur

Lorsque vous découvrez une erreur dans un programme, l'écran est souvent sous SCREEN 0,0,0 (loi de l'ennui maximum).

SCREEN 7 est à taper en frappe aveugle
CONSOLE 0,24 peut éventuellement remettre en place l'écran:

On doit alors retrouver un message d'erreur : Error 2 line 127 ou SN Error In 127

Taper LIST. sur TO7 ou interroger en mode direct: ? ERR ; ERL

Attention, RESUME peut avoir modifié ces paramètres.

Si vous ne découvrez pas la raison de l'erreur, supprimer du programme les SCREEN 0,0,0 , les ON ERROR et contrôlez le CLEAR.

Vérifier les branchements avec RENUM 60000,6000.

Tentez une nouvelle exécution, en utilisant éventuellement TRON et bon courage.

XIII) FICHIERS SÉQUENTIELS

Le programme utilise-t-il des fichiers ? instructions :

```
OPEN "O" , #1, "XXXXXX. DAT"
INPUT #1,....
CLOSE
```

Ces fichiers de données peuvent être recopiés avec de petits programmes comme CASDIS.

Lorsqu'ils sont courts, on a intérêt à les introduire dans le programme sous forme de DATA, en remplaçant les INPUT#1, ... par des READ... Parfois les noms de fichiers sont précédés de "CASS:". Supprimer ce préfixe pour pouvoir utiliser des disquettes.

Certains programmes utilisent plusieurs fichiers de données sous le même nom. En version disquette il faut les numéroter et modifier le programme basic pour laisser le choix du fichier de travail.

- Travaux pratiques 1 : conjugaison I.P.T.

La cassette contient :

ENTCONJ.BAS : programme ENTETE, destiné à faire patienter pendant le chargement du magnétophone, peut être supprimé.

CONJUG.BAS programme principal puis trois fichiers : FICONJ.DAT.

Le programme CONJUG contient la ligne suivante pour la lecture des fichiers :

```
3 IFFO=ITHENF0=0:LOCATE4,23:PRINT"Je mets les phrases en
   mémoire ":OPEN" I",#1,"FICONJ":GOTO4ELSE6
```

La remplacer par :

```
3 PRINT"Sur quel fichier voulez-vous travailler 1,2 ou 3";
   :R$=I NPUT$(1):IF R$>"0" AND R$<"4" THEN OPEN"
   O",#1,"FICONJ"+R$ ELSE 3
```

Puis exécuter 3 fois le programme suivant CASDIS en nommant les fichiers FICONJ1,FICONJ2 et FICONJ3.

```
10 CLS:PRINT"TRANSFERT DE FICHIERS DE DONNEES"
20 PRINT"Source : cassette (magnétophone en position lecture)
30 INPUT but : nom du fichier Disquette ";NOM$
40 OPEN" I", #1, "CASS:
50 OPEN" O", #2, NOM
60 LINEINPUT #1,A$
70 WRITE #2,A$
80 IF EOF<1) THEN CLOSE ELSE 60
```

A la fin de la cassette on trouve : EDCONJUG.BAS programme de création de fichiers.

CAAV.TO7 retour à la mire, peut être supprimé. (exemple de programme compactifié. Une seule ligne de 453 caractères, seul le début de la ligne est listable.)

- Travaux pratiques 2 : Invasion des chiffres

Transformation d'un petit fichier de données en DATA :

La cassette contient :

ENTINV.BAS qui peut être supprimé,

INVAS.BAS programme principal,

puis un mini fichier :

INSINV.DAT dont le seul but est de vous faire jongler avec les touches du magnétophone et de donner du travail aux pirates (Ce fichier n'existait pas dans les premières versions IREM du programme. Il a été introduit pour I.P.T.).

Nous pouvons exécuter LECTDATA permettant de lire les données et de les afficher à l'écran.

```
0
1 CLS:PRINT"lecture de fichier de DONNÉES et écriture de DATA"

10 PRINT"Source:cassette(magnétophone en position lecture )
20 OPEN" I",#1,"CASS:"
30 A=20000:PRINT A"DATA";.
40 INPUT#1,D$
50 PRINT D$;
60 IF EOF(1) THEN CLOSE: END
70 IF POS(0)>35 THEN A=A+l0:PRINT:PRINT A"DATA" ELSE PRINT
80 GOTO 40
```

Nous obtenons les lignes suivantes que l'on conserve dans la mémoire écran :

```
20000 DATA 100, 11, 99, 11, 99, 11, 99
20010 DATA 11, 99, 2, 9, 2, 9, 2, 9, 2
20020 DATA 9, 0, 0, 0
```

Nous chargeons le programme INVAS (listable) `LOAD"INVAS"`

Puis validons les lignes de `DATA`: remonter le curseur et taper trois fois ENTREE.

Dans le programme principal nous trouvons le sous-programme de lecture de fichier :

```
5925 CLS:SCREEN1,4,6:ATTRBO,O:LOCATE9,10:PRINT'Chargement des
   données"
5930 OPEN" I",#1,"I NS INV" 5940 INPUT#1,ZMX
5950 FOR NIV=1TO4
5955 INPUT#1,PNI(NIV),PNS(NIV),DNI(NIV),DNS(NIV)
5957 NEXT NIV
5958 INPUT#1,RET(1),SOM(1),DFMD(2) 5960 CLOSE #1
```

Il reste à modifier le programme INVAS: (transformation des INPUT#1 en READ)

```
5925 CLS:SCREEN1,4,6:ATTRBO,0
5930 RESTORE 20000
5940 READ ZMX
5950 FOR NIV=1TO4
5955 READ PNI(NIV),PNS(NIV),DNI(NIV),DNS(NIV)
5957 NEXT NIV
5958 READ RET(1),SOM(1),DFMD(2)
5960 'à supprimer
SAVE"INVAS" ; les vicieux rajoutent P
```

Le contenu de la cassette peut donc être réduit à ce seul programme qui est alors utilisable avec le Picoréseau.

Lorsque la liste des DATA dépasse une page écran on peut créer un fichier programme à MERGEr dans le programme principal en modifiant LECTDATA

```
25 OPEN "O",#2,"LIGNES.BAS"
30 A=20000:L$=STR$(A)+"DATA"
50 L$=L$+D$
60 IF EOF(1) THEN PRINT #2,L$:CLOSE:END
70 IF LEN(L$)>230 THEN PRINT #2,L$:A=A+10:L$=STR$(A)+"DATA"
   ELSE L$=L$+","
```

Puis RUN
LOAD"PROG"
et MERGE"LIGNES"
enfin SAVE" PROG"

- Travaux pratiques 3 : Promenade

Pour utiliser les disquettes recopier le fichier de donnée (RUN 60050 sous-programme se plaçant à la fin de PROMENAD).

```
60050 OPEN"I",#1,"CASS:INSPRO"
60060 INPUT#1,SENS
60070 INPUT#1,NIV3
60080 INPUT#1,CHIF
60090 INPUT#1,PPP
60100 CLOSE
60110 GOSUB 7580' utilisation de la partie enregistrement du
   programme de modifications accessible par INS + ENTRÉE.
```

La section chargement des données peut être modifiée :

```
7610 CLS:SCREEN1,0,0:LOCATE10,11:ATTRB0,0:PRINT"Chargement des
   données"
7615 'GOTO 7595' Si pas de fichier, magnétophone ou picoréseau
7620 OPEN" I",#1,"INSPRO" 7621 INPUT#1,SENS
7622 INPUT#1,NIV3
7623 INPUT#1,CHIF
7624 INPUT#1,PPP
7630 CLOSE:RETURN
7640 SENS=0:NIV3=0:CHIF=0:PPP=0:RETURN
```

Le ON ERROR 20000 permet de gérer l'utilisation du fichier INSPRO

20000 IF ERL=7620 THEN RÉSUMÉ 7640' DATA remplaçant le fichier

```
20010 PRINT"Erreur"ERR"ligne"ERL:END
```

Un bon exercice de programmation consiste à supprimer complètement ce fichier et à transformer le sous-programme de modifications de données (lignes 7360-7590) en un programme les implantant directement en mémoire, ligne 7640 : suivre le chaînage à partir de 611C ou 2113 puis POKEr les nombres au format de STR$( ).

Les GR$ sont rangés dans un fichier de DAT qui est assez long à lire et qui est chargé à chaque exécution du programme par la ligne 14:

```
14 OPEN"I",#1,"PROMGR":FORT=0TO44:FORJ=1TO8:INPUT#1,A(J):NEXTJ:D
EFGR$(I)=A(1),A(2),A(3),A(4),A(5),A(6),A(7),A(8):NEXT I:CLOSE
```

Il possible de supprimer ce fichier et d'implanter directement les GR$ en mémoire

```
30000 DEFGR$(0)=....
.....
30440•DEFGR$(44)=...
```

ces lignes sont crées sous forme fichier ASCII :

```
60000 OPEN" I",#1,"CASS:PROMGR":OPEN"O",#2, "LIGNÉS. BAS"
60010 FOR I=0 TO 44:PRINT#2,STR$(30000+I*10);
   "DEFGR$("MID$(STR$( I),2,LEN(STR$(I>>-1)")=";
60020 FOR J=1TO 8:INPUT#1,A:
   PRINT#2,MID$(STR$(A),2,LEN(STR$(A))-1);
60030 IF J<8 THEN PRINT#2,",";
60040 NEXT J:PRINT#2:NEXT:CLOSE
60042 MERGE"LIGNES"
60044 KILL"LIGNES.BAS
```

et remplacer la ligne 14 par : `14 GOSUB 30000` puis terminer avec : `30450 RETURN`

En système disquette, il est performant d'utiliser un fichier binaire PRONGR.BIN à créer sur TO7-70, par exemple, après le lancement de PROMENAD par :

```
SAVEM"PRONGR",&HDFFF-8*45,&HDFFF,0
```

ou plus précisément :

```
IF PEEK(0)=32 THEN USERAF=&H602D ELSE USERAF=&H2070
ADRDEB=PEEK(USERAF)*256+PEEK(USERAF+1)
N=45:ADRFIN=APRDEB+N*8-1
SAVEM"PROMGR",
```

On peut retrouver les `GR$` par un simple `14 LOADM" PROMGR"`

Si l'on veut tenir compte des décalages entre les micros : `14 GOSUB 30000`

```
30000 IF PEEK(0)=32 THEN USERAF=&H602D ELSE USERAF=&H2070
30010 ADRDEB=PEEK(USERAF)*256+PEEK(USERAF+1)
30020 LOADM"PROMGR",ADRDEB-&HDE97
30030 RETURN
```

XIV ACTE 3 : FICHIERS BINAIRES

Souvent les programmes machines ne sont pas compatibles sauf s'il sont parfaitement translatables et si la correspondance des adresses a été trouvée. C'est souvent difficile car les moniteurs TO7 et MO5 n'ont pas la même logique sauts vectorisés sur TO7-TO9, interruptions sur MO5.

Copier

Pour dupliquer des fichiers binaires sur disquettes, le plus simple est d'utiliser l'ordre `COPY` :

```
COPY "FICH.BIN" avec un seul lecteur de disquette.
COPY "0:FICH.BIN" TO "1:FICH.BIN" avec deux lecteurs de
   disquette Thomson.
COPY A:FICH.BIN B: sur le serveur MS-DOS du nanoréseau.
```

LOADM-SAVEM

Pour une copie utilisant une cassette il faut charger le fichier en mémoire avec l'instruction `LOADM` et le sauver avec `SAVEM`.

Ces deux ordres ne sont pas symétriques. On peut facilement charger mais pour effectuer un LOADM il faut connaître l'adresse du début, celle de la fin et l'adresse de lancement du programme binaire.

L'adresse d'exécution se trouve facilement. Elle est rangée en &H223F sur MO5 et en &H623F sur TO7. Par exemple sur TO7 : ADEXEC=256*PEEK(&H623F)+PEEK(&H6240).

On la trouve aussi à la fin de l'enregistrement : les cinq derniers octets sont FF, 00, 00, ADEXECI ADEXEC2 suivis du reste du buffer précédent.

L'adresse de début est rangée en &H44EF sur un MO5 du nanoréseau sinon deux possibilités pour trouver les deux dernières adresses

- Première méthode

On peut DUMPer la cassette ou la disquette (voir programme DUMPCASS ou SCRUTCAS Théophile).

Le premier octet est 00 suivi de deux octets indiquant la longueur de l'enregistrement suivis de deux autres octets donnant l'adresse du début. Soit après une suite de FF,FF,FF indiquant le début de la cassette on trouve: 00,LG1,LG2,ADDEBI,ADDEB2,.. puis le programme binaire.

L'adresse de fin est alors: ADFIN=ADDEB+LG-1.

- Deuxième méthode

Pour trouver ADDEB et ADFIN on peut aussi vider la mémoire puis en tester le début et la fin: Par exemple si l'on a trouvé un CLEAR &HA000 dans le programme chargeur BASIC (TO7-70) on peut lancer le programme

```
10 DEB=&HA000
20 CLEAR DEB 3
0 DEB=&HA000:FIN=&HDFFF'Reécrire DEB effacé par CLEAR
40 FOR I=DEB TO FIN:POKE I,0:NEXT
50 LOADM"CASS:"
60 FOR I=DEB TO FIN
70 IF PEEK(I»0 THEN ADDEB=I: I=FIN
80 NEXT
90 PRINT"Adresse de début &H";HEX$(ADEB)
100 FOR I=FIN TO DEB STEP -1
```

```
110 IF PEEK(I»O THEN ADFIN=I:I=DEB
120 NEXT
130 PRINT"Adresse de fin &H";HEX$(ADFIN)
140 ADEXEC=PEEK(&H623F)*256+PEEK(&H6240)
150 PRINT"Adresse d'exécution &H";HEX$(ADEXEC)
```

Avant l'enregistrement du programme binaire on notera dans le chargeur basic ces trois adresses pour éviter de recommencer ce travail.

Un certain nombre de fichiers binaires sont directement exécutables. LOADM met à un le drapeau d'EXEC et lance le programme: On peut les charger avec un décalage et garder la main au basic.

```
LOADM"CASS:",&H1000
```

Ne pas oublier ce décalage dans les calculs.

- Travaux pratiques 4 : Tennis

Dans le programme basic ENTETE on trouve :

```
POKE &H61A2,&HFF remise à vrai du drapeau de protection, anti
   "E XEC 1218"
POKE &H6294,&H32,&H62  LEAS
POKE &H6296,&H39 RTS
```

Modification de l'index de la page 0 du système. Supprime l'action des touches STOP et CNT C (à connaître).

CLEAR ,&H7500 obligatoire sinon la saturation du haut de la moire plante l'ordinateur à la fin de l'enregistrement. (remarquer que l'on ne peut utiliser le DOS ordinaire.)

Puis LOADM pour charger le programme binaire TEN70.BIN.

Le début de l'enregistrement TEN70 est: 00,5E,BO,81,50,20,24,:. L'adresse de début est donc &H8150.

L'adresse de fin est &H8150+&H5EBO-1=&HDFFF.

Le chargeur BASIC TENNIS s'écrit :

```
10 CLS:PRINT"Patience je charge TENNIS pour TO 7-70"
20 POKE &H6294,&H32
30 POKE &HG295,&H62
40 POKE &H6296,&H39
50 CLEAR ,&H7500
60 PRINT CHR$(20>'curseur invisible
```

```
70 LOADM "TEN70", ,R
80 'SAVEM "TELAT70",&H8150,&HDFFF,&H9500'pour mémoire
```

Puis changer de bande

```
SAVE "TENNIS"
SAVEM "TENT70",&H8150,&HDFFF,&H9500
```

Pour TO7+16k

```
SAVEM "TENTO7",&H7950,&HDFFF,&H7950
```

Pour MO5 : pas de POKE dans le chargeur.

```
CLEAR ,&H4000
SAVEM "TENMO5",&H4150,&HDFFF,&H8500
```

CONCLUSION

En espérant que ces quelques trucs plus ou moins faciles vous aiderons à comprendre le système. Bien se rappeler que ce qui fait l'intérêt d'un programme est sa documentation. Ne piratez pas : Il faut que l'on trouve des logiciels pas chers et de qualité et cesser de payer implicitement pour toutes les copies pirates.