

## 9/8.9

# Instruction CAT évoluée

---

Aux vues des insuffisances de la commande Basic CAT (listage de répertoires), nous avons créé une extension de cette commande permettant :

- le tri par ordre alphabétique, inverse alphabétique, par taille de fichiers et par extension,
- la sortie du catalogue trié sur l'écran, sur une imprimante ou dans un fichier.

Le programme développé ici est écrit en Basic et en Assembleur.

La syntaxe générale d'une commande CAT évoluée est la suivante :

**<Specification fichier> , <Type de tri> , <Organe de sortie>**

- **<Specification fichier>**

permet de définir le ou les fichiers sur le(s)quel(s) porte(nt) la commande. Cette spécification est du type **<Nom> . <Extension>** avec les restrictions suivantes :

<b>&lt;Nom&gt;</b>	est un nom de fichier CP/M valide comportant entre 1 et 8 caractères (y compris le joker*),
<b>&lt;Extension&gt;</b>	est une extension CP/M valide comportant obligatoirement 3 caractères (même espace).

- **<Type de tri>** est une des lettres suivantes, indifféremment majuscule ou minuscule :

<b>A</b>	- - - tri alphabétique,
<b>I</b>	- - - tri inverse alphabétique,
<b>T</b>	- - - tri par taille de fichier,
<b>E</b>	- - - tri par extension de fichier.

- **<Organe de sortie>** est une des lettres suivantes, indifféremment majuscule ou minuscule :

<b>E</b>	- - - sortie écran,
<b>I</b>	- - - sortie imprimante,
<b>F</b>	- - - écriture dans un fichier.

Lorsque l'option F apparaît dans la commande, elle doit être suivie d'un nom de fichier CP/M valide (1 à 8 caractères pour le radical et 0 à 3 caractères pour l'extension).

*Exemples :*

Visualisation de tous les fichiers par ordre alphabétique : \*.\* ,A,E

```

CAT *.* ,A,E
3DEX          BIN          1 KOctet(s).
ADJ           BAS          1 KOctet(s).
ADJ1          BIN          17 KOctet(s).
ADJANI2       BIN          1 KOctet(s).
AF2D          2 KOctet(s).
AF3D          2 KOctet(s).
AS2           2 KOctet(s).
ASM           2 KOctet(s).
BIN2          BIN          17 KOctet(s).
COMPAC2       BAS          4 KOctet(s).
DAGRAPH       BIN          17 KOctet(s).
DEBUG         BAS          5 KOctet(s).
DECOM2        2 KOctet(s).
DEFIN         1 KOctet(s).
ESSS          BIN          1 KOctet(s).
F1            BIN          2 KOctet(s).
GEFOR         BAS          3 KOctet(s).
INIT          BAS          1 KOctet(s).
MAXSUI        4 KOctet(s).
MUSE          BIN          1 KOctet(s).
P2            BIN          5 KOctet(s).
PRES          BIN          17 KOctet(s).
REG           4 KOctet(s).
STAT          DOC          1 KOctet(s).
SUPERCOM      BAS          4 KOctet(s).
T2            BIN          6 KOctet(s).
TAF2D         1 KOctet(s).
X             8 KOctet(s).
X2            3 KOctet(s).
ZARKA         BAS          3 KOctet(s).

```

Impression des fichiers dont le préfixe commence par AD, par extension de fichier : AD\*.\* ,E,I

```

CAT AD*.* ,E,I
ADJ           BAS          1 KOctet(s).
ADJ1          BIN          17 KOctet(s).
ADJANI2       BIN          1 KOctet(s).

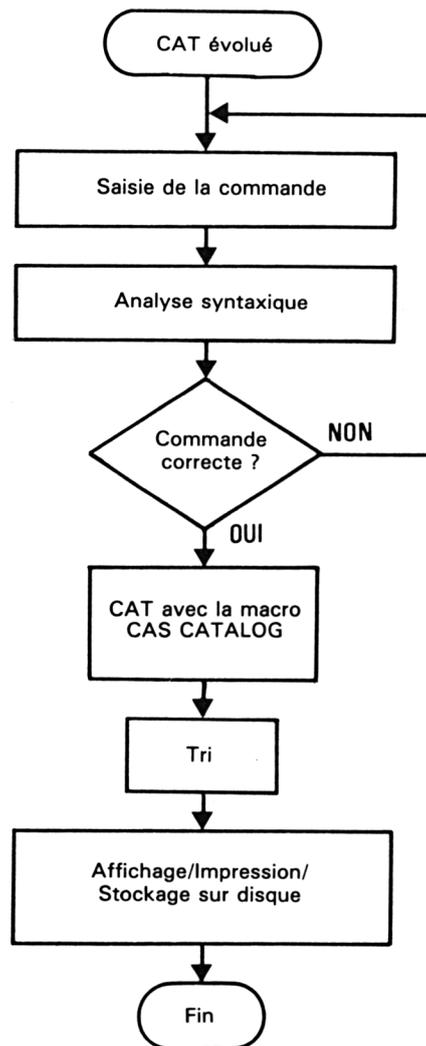
```

Stockage sur disque des fichiers d'extension nulle, par taille de fichier, dans le fichier stat.doc : \*. ,T,F,stat.doc

CAT *. ,T,F,STAT.DOC	
DEFIN	1 KOctet(s).
TAF2D	1 KOctet(s).
AF2D	2 KOctet(s).
AF3D	2 KOctet(s).
AS2	2 KOctet(s).
ASM	2 KOctet(s).
DECOM2	2 KOctet(s).
X2	3 KOctet(s).
MAXSUI	4 KOctet(s).
REG	4 KOctet(s).
X	8 KOctet(s).

## Analyse du programme

La logique du programme respecte l'ordinogramme suivant :

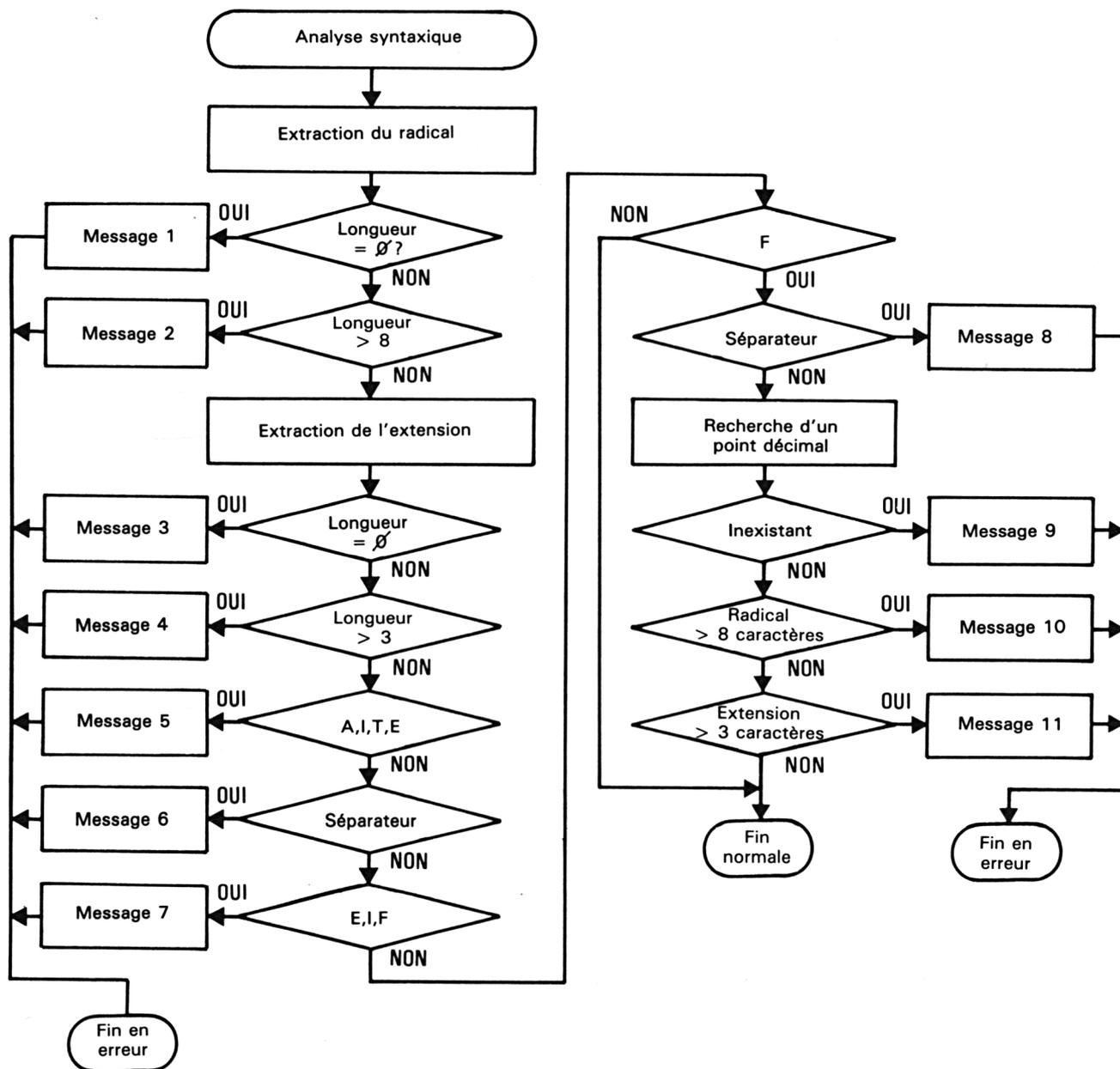


L'analyse syntaxique s'assure que la commande entrée pourra être traitée par le logiciel :

- spécification de fichier correcte,
- type de tri correct,
- organe de sortie correct,
- spécification de fichier correcte pour l'organe de sortie F.

Si un des points listé ci-dessus n'est pas conforme au format attendu, un message d'erreur est affiché, la commande entrée est ignorée, et le programme réexécuté pour saisir une nouvelle commande.

La logique de l'analyse syntaxique respecte l'ordinogramme suivant :



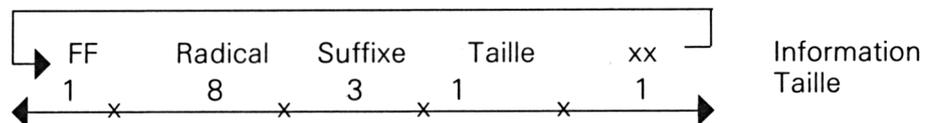
Le programme Basic utilise une routine du Firmware : **CAS CATALOG**. Cette routine permet de lire le catalogue d'une disquette.

```

1          ORG 9000H
2          LOAD 9000H
3          ;-----
4          CASCAT: EQU 0BC9BH          ;CAS CATALOG
5          ;-----
6 9000 1100B0          LD DE,8000H          ;à Buffer
7 9003 CD9BBC          CALL CASCAT          ;Catalogue
8 9006 C9             RET
9          END

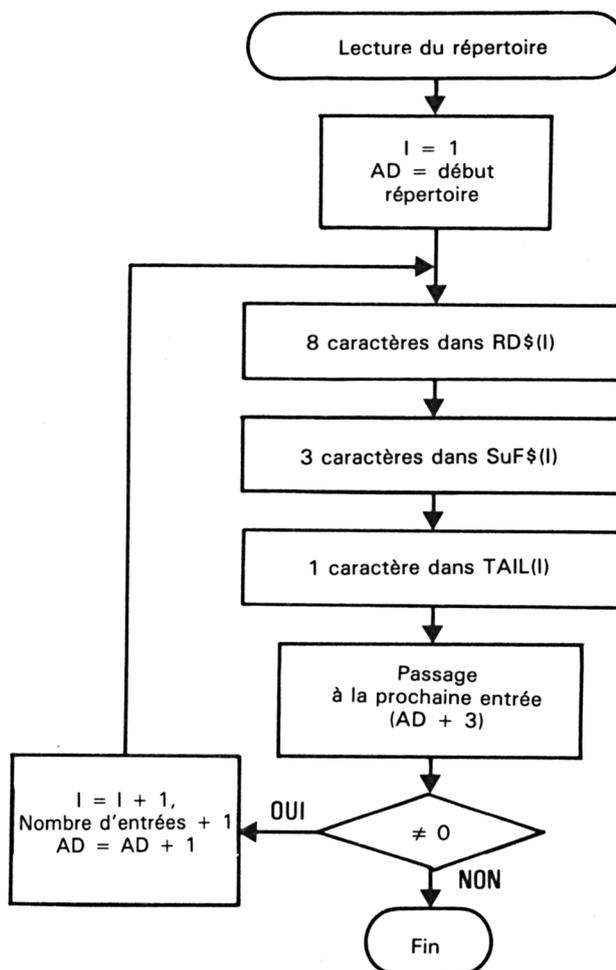
```

Ce catalogue est affiché sur l'écran et stocké en mémoire RAM à une adresse quelconque selon le format suivant :



Le délimiteur &HFF est utilisé entre deux entrées de fichiers et le délimiteur &H00 marque la fin de la dernière entrée dans le répertoire.

La lecture des nom, suffixe et taille des fichiers d'un répertoire suit les étapes de l'ordinogramme suivant :



Le listing du programme est le suivant :

```

1000 '=====
1010 'Instruction CAT evoluee
1020 '=====
1030 '-----
1040 ' Initialisations
1050 '-----
1060 DIM RD$(30),SUF$(30),TAIL(30),TP(30)
1070 DIM TT1$(30),TT2$(30),TT3(30)
1080 '
1090 FOR I=&9000 TO &9006
1100 READ A$
1110 A=VAL("&H"+A$)
1120 POKE I,A
  
```

```
1130 NEXT I
1140 '
1150 DATA 11,00,80,CD,9B,BC,C9
1160 '
1170 '-----
1180 ' Saisie de la chaine de commande
1190 '-----
1200 CLS
1210 PRINT "Instruction CAT evoluee"
1220 PRINT "-----"
1230 PRINT
1240 LINE INPUT "Chaine de commande :";C$
1250 C$=UPPER$(C$)
1260 '
1270 '-----
1280 ' Analyse syntaxique
1290 '-----
1300 P1=INSTR(C$,".")
1310 IF P1=0 THEN E=1:GOTO 2890
1320 IF P1>9 THEN E=2:GOTO 2890
1330 C1$=LEFT$(C$,P1-1)
1340 D$=RIGHT$(C$,LEN(C$)-P1)
1350 P2=INSTR(D$,"")
1360 IF P2=0 THEN E=3: GOTO 2890
1370 IF P2>4 THEN E=4:GOTO 2890
1380 C2$=LEFT$(D$,P2-1)
1390 D$=RIGHT$(D$,LEN(D$)-P2)
1400 E$=LEFT$(D$,1)
1410 IF E$<>"A" AND E$<>"I" AND E$<>"T" AND E$<>"E" THEN E=5:GOTO 2890
1420 P3=INSTR(D$,"")
1430 IF P3<>2 THEN E=6:GOTO 2890
1440 C3$=LEFT$(D$,1)
1450 D$=RIGHT$(D$,LEN(D$)-2)
1460 E$=LEFT$(D$,1)
1470 IF E$<>"E" AND E$<>"I" AND E$<>"F" THEN E=7:GOTO 2890
1480 C4$=LEFT$(D$,1)
1490 IF E$<>"F" THEN 1610
1500 P4=INSTR(D$,"")
1510 IF P4=0 THEN E=8:GOTO 2890
1520 D$=RIGHT$(D$,LEN(D$)-2)
1530 P5=INSTR(D$,".")
1540 IF P5=0 THEN E=9:GOTO 2890
1550 IF P5>9 THEN E=10:GOTO 2890
1560 C5$=LEFT$(D$,P5-1)
1570 D$=RIGHT$(D$,LEN(D$)-P5)
1580 IF LEN(D$)>3 THEN E=11:GOTO 2890
1590 C6$=D$
1600 '
1610 '-----
1620 'La chaine de commande est correcte
1630 '-----
1640 '
1650 '-----
1660 'Lecture du repertoire
1670 '-----
```

```

1680 INK 0,0:PEN 0
1690 CALL &9000 'Appel du S/P Assembleur
1700 CLS:PEN 1
1710 PRINT"Tri en cours...":PRINT
1720 '
1730 '-----
1740 'Classement des entrees
1750 '-----
1760 AD=&8000
1770 AD=AD+1
1780 I=1:NB=1          'Initialisation
1790 FOR J=1 TO 8
1800   RD$(I)=RD$(I)+CHR$(PEEK(AD))
1810   AD=AD+1
1820 NEXT J
1830 FOR J=1 TO 3
1840   SUF$(I)=SUF$(I)+CHR$(PEEK(AD))
1850   AD=AD+1
1860 NEXT J
1870 TAIL(I)=PEEK(AD)
1880 AD=AD+2 'Passage a l'entree suivante
1890 IF PEEK(AD)<>0 THEN I=I+1:AD=AD+1:NB=NB+1:GOTO 1790
1900 '
1910 '-----
1920 'Toutes les entrees sont memorisees
1930 '-----
1940 IF C3$="A" THEN 2580
1950 '
1960 '-----
1970 'Classement par ordre inverse alphabetique
1980 '-----
1990 IF C3$<>"I" THEN 2150
2000 FOR I=1 TO NB
2010   TT1$(I)=RD$(NB-I+1)
2020   TT2$(I)=SUF$(NB-I+1)
2030   TT3(I)=TAIL(NB-I+1)
2040 NEXT I
2050 FOR I=1 TO NB
2060   RD$(I)=TT1$(I)
2070   SUF$(I)=TT2$(I)
2080   TAIL(I)=TT3(I)
2090 NEXT I
2100 GOTO 2580
2110 '
2120 '-----
2130 'Classement par taille
2140 '-----
2150 IF C3$<>"T" THEN 2390
2160 FOR I=1 TO NB
2170   TP(I)=I
2180 NEXT I
2190 V=0
2200 FOR I=1 TO NB-1
2210   IF TAIL(I)<=TAIL(I+1) THEN 2240
2220   A=TAIL(I+1):TAIL(I+1)=TAIL(I):TAIL(I)=A:V=1
2230   A=TP(I+1):TP(I+1)=TP(I):TP(I)=A

```

```

2240 NEXT I
2250 IF V=1 THEN 2190
2260 FOR I=1 TO NB
2270   TT1$(I)=RD$(TP(I))
2280   TT2$(I)=SUF$(TP(I))
2290 NEXT I
2300 FOR I=1 TO NB
2310   RD$(I)=TT1$(I)
2320   SUF$(I)=TT2$(I)
2330 NEXT I
2340 GOTO 2580
2350 '
2360 '-----
2370 'Classement par extension
2380 '-----
2390 FOR I=1 TO NB
2400   TP(I)=I
2410 NEXT I
2420 V=0
2430 FOR I=1 TO NB-1
2440   IF SUF$(I)<=SUF$(I+1) THEN 2470
2450   A$=SUF$(I+1):SUF$(I+1)=SUF$(I):SUF$(I)=A$:V=1
2460   A=TP(I+1):TP(I+1)=TP(I):TP(I)=A
2470 NEXT I
2480 IF V=1 THEN 2420
2490 FOR I=1 TO NB
2500   TT1$(I)=RD$(TP(I))
2510   TT3(I)=TAIL(TP(I))
2520 NEXT I
2530 FOR I=1 TO NB
2540   RD$(I)=TT1$(I)
2550   TAIL(I)=TT3(I)
2560 NEXT I
2570 '
2580 '-----
2590 'Toutes les entrees sont classees
2600 '-----
2610 CLS
2620 IF C4$<>"F" THEN 2650
2630 A#=C5#+". "+C6$:OPENOUT A#
2640 A#="CAT "+C$:PRINT#9,A#
2650 IF C4$="E" THEN PRINT"CAT "+C#
2660 IF C4$="I" THEN PRINT#8,"CAT "+C#
2670 FOR I=1 TO NB
2680   OK=0
2690   IF INSTR(C1$,"*")<>0 AND LEFT$(C1$,1)<>"*" THEN 2720
2700   IF INSTR(C1$,"*")<>0 AND LEFT$(C1$,1)="*" THEN OK=1:GOTO 2750
2710   IF C1#=RD$(I) THEN OK=1:GOTO 2750 ELSE OK=0:GOTO 2820
2720   P=INSTR(C1$,"*")
2730   IF LEFT$(C1$,P-1)=LEFT$(RD$(I),P-1) THEN OK=1
2740   '
2750   IF OK<>1 THEN 2860
2760   IF INSTR(C2$,"*")<>0 AND LEFT$(C2$,1)<>"*" THEN 2790
2770   IF INSTR(C2$,"*")<>0 AND LEFT$(C2$,1)="*" THEN OK=2:GOTO 2820
2780   IF C2#=SUF$(I) THEN OK=2:GOTO 2820 ELSE OK=0:GOTO 2820
2790   P=INSTR(C2$,"*")

```

```

2800 IF LEFT$(C2$,P-1)=LEFT$(SUF$(I),P-1) THEN OK=2
2810 '
2820 IF OK<>2 THEN 2860
2830 IF C4$="E" THEN PRINT RD$(I),SUF$(I),TAIL(I);"KOctet(s). "
2840 IF C4$="I" THEN PRINT#8,RD$(I),SUF$(I),TAIL(I);"KOctet(s). "
2850 IF C4$="F" THEN PRINT#9,RD$(I),SUF$(I),TAIL(I);"KOctet(s). "
2860 NEXT I
2870 IF C4$="F" THEN CLOSEOUT
2880 END
2890 '-----
2900 'Erreur pendant l'analyse syntaxique
2910 '-----
2920 IF E=1 THEN PRINT"Point decimal attendu"
2930 IF E=2 THEN PRINT"Longueur prefixe incorrecte"
2940 IF E=3 THEN PRINT"Virgule attendue"
2950 IF E=4 THEN PRINT"Longueur suffixe incorrecte"
2960 IF E=5 THEN PRINT"A,I,T ou E attendu"
2970 IF E=6 THEN PRINT"Virgule attendue"
2980 IF E=7 THEN PRINT"E, I ou F attendu"
2990 IF E=8 THEN PRINT"Virgule attendue"
3000 IF E=9 THEN PRINT"Point decimal attendu"
3010 IF E=10 THEN PRINT"Longueur prefixe incorrecte"
3020 IF E=11 THEN PRINT"Longueur suffixe incorrecte"
3030 PRINT:PRINT"Appuyez sur une touche..."
3040 A$=INKEY$:IF A$="" THEN 3040
3050 RUN

```

- Lignes 1060 et 1070... Déclaration des tableaux de travail
- Lignes 1090 à 1150... Mise en mémoire du programme  
Assembleur
- Lignes 1200 à 1250... Saisie de la chaîne de commande CAT
- Lignes 1300 à 1590... Analyse syntaxique  
Extraction :
  - préfixe (ligne 1330),
  - suffixe (ligne 1380),
  - type de tri (ligne 1440),
  - organe de sortie (ligne 1480),
  - préfixe option T (ligne 1560),
  - suffixe option T (ligne 1590),
- Lignes 1680 à 1710... Lecture du répertoire en utilisant la macro  
**CAS CATALOG** du firmware
- Lignes 1760 à 1890... Mémorisation dans les tableaux des don-  
nées du répertoire (noms, suffixes et tailles)
- Ligne 1940... Aucune opération si tri alphabétique
- Lignes 1990 à 2100... Tri par ordre inverse alphabétique
- Lignes 2150 à 2340... Tri par taille de fichier
- Lignes 2390 à 2560... Tri par extension
- Lignes 2610 à 2880... Affichage, impression ou stockage sur dis-  
que du résultat du tri
- Lignes 2920 à 3050... Erreur pendant l'analyse syntaxique

