

# **SOLAR**

EDITEX

**Correction/édition  
de programmes symboliques**

ÉDITEX

MANUEL DE RÉFÉRENCE

SOMMAIRE	Pages
1 - PRÉSENTATION D'ÉDITEX	1
2 - INTRODUCTION A ÉDITEX	2
2.1 - GÉNÉRALITÉS	2
2.2 - PRINCIPES ET SYNTAXE	2
2.3 - NOTATIONS	2
3 - DESCRIPTION DES COMMANDES	3
3.1 - REPÉRAGE DE LIGNES	3
3.2 - UNITÉS SYMBOLIQUES UTILISÉES PAR ÉDITEX	4
3.3 - COMMANDES COPY ET DELETE	5
3.4 - COMMANDE INSERT	6
3.5 - COMMANDE REPLACE	6
3.6 - COMMANDE EXCHANGE	7
3.7 - COMMANDES DE TABULATION : TABS ET TABC	7
3.8 - COMMANDE BLANK	8
3.9 - COMMANDE SEPARATOR	8
3.10 COMMANDE EOT	8
3.11 EXEMPLE D'UTILISATION DES COMMANDES	9
4 - UTILISATION DE L'ÉDITEUR ÉDITEX	11
4.1 - ACTIVATION DE L'ÉDITEUR	11
4.1.1 - COMMANDE ITEX	11
4.1.2 - COMMANDE DTEX	11
4.1.3 - COMMANDE CTEX	11
4.2 - MESSAGES ÉMIS PAR EDITEX	11
4.3 - CORRECTION DES ERREURS	12
4.3.1 - LES ERREURS DE PARITÉ	12
4.3.2 - ERREURS DE SYNTAXE DANS LES COMMANDES	12
4.3.3 - ERREURS LIÉES A L'EXÉCUTION DES COMMANDES	12
4.4 - INTERRUPTION D'UNE ÉDITION EN COURS	12
ANNEXE	13
A - SYNOPTIQUE DES COMMANDES	13
B - LISTE DES NUMÉROS D'ERREURS	15

## 1 - PRÉSENTATION D'EDITEX

EDITEX est un programme de génération et de correction de textes écrits dans un langage quelconque.

On l'utilise pour :

- la correction et la modification des programmes.

Il est souvent difficile de modifier un programme enregistré sur un support séquentiel tel que le ruban perforé ou la bande magnétique. EDITEX rend cette tâche très aisée.

- la préparation des programmes.

Les programmes qui se présentent d'abord en langage source (en symbolique ou en langage évolué) doivent être transcrits sur un support que le calculateur peut lire : ruban perforé, bande magnétique, etc...

La préparation se fait avec possibilité de tabulation ce qui permet d'obtenir des listings de programmes correctement présentés.

- le changement de support des programmes.

EDITEX facilite les manipulations du type "cartes à bande" ou l'obtention des listes de programmes.

EDITEX fonctionne :

- en mode conversationnel exclusivement sous le contrôle des systèmes d'exploitation BOS-A, BOS-B et BOS-C,
- en mode conversationnel ou en mode train de travaux (BATCH) sous le contrôle des systèmes BOS-D et BACKM (moniteur permettant une activité Background sous RTES-D).

La configuration minimale nécessaire pour l'utilisation d'EDITEX est la suivante :

- un calculateur SOLAR 16 avec une mémoire de :
  - 4 Kmots en ce qui concerne BOS-A
  - 8 Kmots en ce qui concerne BOS-B et BOS-C
  - 16 Kmots en ce qui concerne BOS-D
  - 32 Kmots en ce qui concerne BACKM
- un périphérique de dialogue, téléimprimeur ou console de visualisation,
- un périphérique d'entrée ainsi qu'un périphérique de sortie.

## 2 - INTRODUCTION A EDITEX

### 2.1 - GÉNÉRALITÉS

L'utilisateur dirige le fonctionnement d'EDITEX au moyen de commandes dont on trouvera la liste dans le chapitre suivant.

En cours de travail, il reçoit d'EDITEX, des messages d'erreur ou de dialogue.

### 2.2 - PRINCIPES ET SYNTAXE

Les traitements effectués par EDITEX se font au niveau de la ligne, qui est définie comme une suite de caractères terminée par un  $\text{\textcircled{CR}}$  (Carriage - Return ou retour chariot).

Les lignes du texte source sont repérées par recherche de contexte ou par déplacement relatif à la ligne courante.

Enfin, dans le but d'obtenir un texte qui fournira une liste lisible et bien présentée, EDITEX effectue automatiquement les opérations suivantes :

- suppression dans la ligne de tous les caractères non imprimables.
- génération d'un caractère  $\text{\textcircled{LF}}$  (Line Feed) derrière le retour-chariot  $\text{\textcircled{CR}}$
- suppression dans la ligne de tous les caractères qui précèdent le  $\text{\textcircled{\leftarrow}}$
- suppression dans la ligne du caractère qui précède le  $\text{\textcircled{\uparrow}}$  sauf s'il s'agit de  $\text{\textcircled{\uparrow}}$ ,  $\text{\textcircled{\leftarrow}}$  ou  $\text{\textcircled{CR}}$

Exemple :

Le texte source :

LA MEM1  $\text{\textcircled{CR}}$  AD MEM1  $\text{\textcircled{\uparrow}}$  2  $\text{\textcircled{CR}}$   $\text{\textcircled{LF}}$   
 STB MEM3  $\text{\textcircled{\leftarrow}}$  STA MEM1  $\text{\textcircled{CR}}$

correspond au texte objet :

LA MEM1  $\text{\textcircled{CR}}$   $\text{\textcircled{LF}}$   
 AD MEM2  $\text{\textcircled{CR}}$   $\text{\textcircled{LF}}$   
 STA MEM1  $\text{\textcircled{CR}}$   $\text{\textcircled{LF}}$

### 2.3 - NOTATIONS

Dans la description d'une commande les notations suivantes sont prises :

$[ \dots ]$  Indication d'une option de syntaxe

$\left\{ \begin{array}{l} \dots \\ \dots \end{array} \right\}$  Indication d'un choix de syntaxe

$\left\{ \begin{array}{l} L \\ \underline{NL} \end{array} \right\}$  Le choix souligné est celui qui est pris par défaut

### 3 - DESCRIPTION DES COMMANDES

#### 3.1 - REPÉRAGE DE LIGNES

Définitions :

**Ligne courante** : c'est la dernière ligne insérée dans le texte objet ou la dernière ligne lue dans le texte source mais non traitée par la commande précédente.

Dans certains cas (état initial, insertion vide...) elle n'existe pas.

Toutes les commandes commencent par traiter cette ligne ou à défaut la prochaine à lire dans le texte source.

**Ligne pointée** : c'est la ligne du texte source jusqu'à laquelle portent certaines commandes.

La description de ces commandes contiendra toujours une définition explicite de la "ligne pointée".

- Les commandes Copy, Delete et Replace portent généralement sur plusieurs lignes.

Leur syntaxe renferme un paramètre p, appelé "paramètre de repérage de ligne" qui sert à déterminer jusqu'où porte la commande dans le texte source et se définit de la façon suivante :

$$p = \left[ \begin{array}{c} / \text{chaîne1} / \\ \left[ \begin{array}{c} + d \\ \underline{0} \end{array} \right] \\ \left[ \begin{array}{c} d \\ \underline{1} \end{array} \right] \end{array} \right]$$

où :

- d est un nombre décimal
- chaîne 1 est une suite de caractères ne comprenant pas le caractère "séparateur de chaîne" qui est implicitement le caractère  $\textcircled{/}$  (slash) mais peut être modifié par commande.

Dans chaîne 1 le nombre d'espaces ( ) d'une suite d'espaces n'est pas significatif.

Exemple : `STA TØTØ`

est équivalent à : `STA TØTØ`

Lorsque p est absent dans la commande, EDITEX le prend par défaut sous la forme  $p = d$  avec  $d = 1$

Ce paramètre autorise trois modes de repérage de ligne :

- repérage par déplacement relatif à la ligne courante :  $p = d$

d est le numéro de la dernière ligne qui sera traitée par la commande. La ligne courante lorsqu'elle existe porte le numéro 1.

- repérage par recherche de contexte :  $p = / \text{chaîne1} /$

Dans ce cas, la commande porte jusqu'à la ligne contenant chaîne 1 (ligne pointée). Après l'exécution de la commande elle devient ligne courante.

-Remarque :

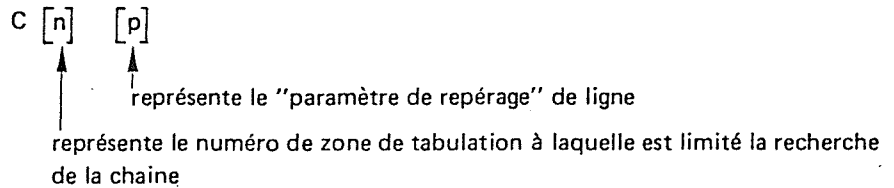
EDITEX accepte l'emploi simultané de ces 2 modes de repérage de ligne :  $p = / \text{chaîne1} / + d$

La commande portera alors dans un premier temps jusqu'à la ligne contenant Chaîne1 puis elle se poursuivra jusqu'à ce que le nombre de lignes indiqué par d soit traité.

- repérage par recherche de contexte zonée :

Lorsque le paramètre de repérage de lignes est exprimé sous la forme d'une chaîne de caractères, on peut limiter la recherche de celle-ci dans les lignes traitées sur une zone de tabulation seulement, en précisant le numéro (n) de celle-ci derrière le caractère représentant le type de la commande.

Exemple :



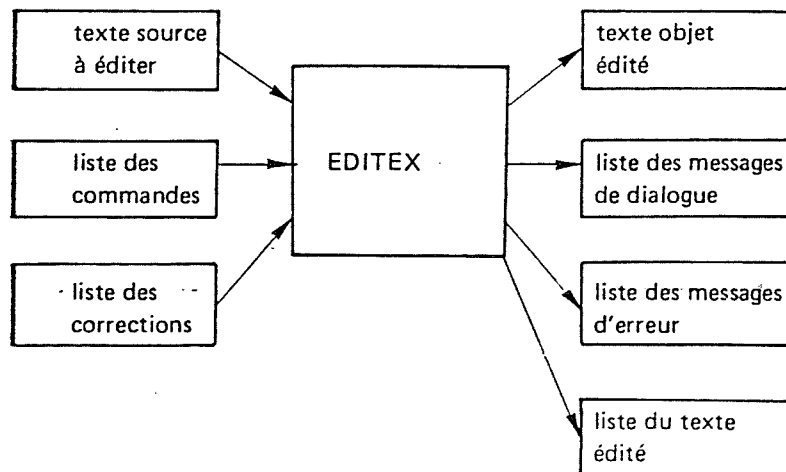
### 3.2 - UNITÉS SYMBOLIQUES UTILISÉES PAR EDITEX

L'éditeur EDITEX est un processeur géré par un superviseur dont le manuel de référence est supposé connu du lecteur.

Il utilise un certain nombre d'unités symboliques pour ses demandes d'entrées-sorties.

Ce sont :

SI	(Symbolic Input)	support du texte source à éditer
PC	(Processor's Command)	entrée des commandes
SO	(Symbolic Output)	support du texte objet édité
EL	(Error Listing)	sorties des messages d'EDITEX
EC	(Error Correction)	entrée des corrections
LO	(List Output)	liste du texte édité



Avant d'activer l'éditeur, l'utilisateur doit affecter à chaque unité symbolique une unité fonctionnelle dont il dispose dans sa configuration.

Exemple d'affectation d'unités fonctionnelles aux unités symboliques nécessaires à EDITEX.

* SI	HR	affectation du lecteur rapide à SI
* SØ	HP	affectation du perforateur rapide à SØ
* PC	TK	] affectation du clavier du téléimprimeur à PC et EC
* EC	TK	
* EL	TS	affectation de la feuille du téléimprimeur à EL
* LØ	LP	affectation de l'imprimante à LØ

### 3.3 - COMMANDES COPY ET DELETE

Syntaxe :  $C [n] [p] \textcircled{\text{CR}}$   
 $D [n] [p] \textcircled{\text{CR}}$  avec p =  $\left[ \begin{array}{l} / \text{chaîne1} / \left[ \begin{array}{l} + d \\ \text{Q} \end{array} \right] \\ \left[ \begin{array}{l} d \\ 1 \end{array} \right] \end{array} \right]$

#### COPY

a) -  $C / \text{chaîne1} / \left[ \begin{array}{l} + d \\ \text{Q} \end{array} \right]$

provoque la recopie du texte source dans le texte objet, depuis la ligne courante incluse (si elle existe) jusqu'à la ligne pointée par le paramètre de repérage exclue.

b) - C d

provoque la recopie de d lignes du texte source dans le texte objet, depuis la ligne courante incluse (si elle existe)

#### DELETE

a) -  $D / \text{chaîne1} / \left[ \begin{array}{l} + d \\ \text{Q} \end{array} \right]$

provoque la suppression dans le texte objet des lignes du texte source, depuis la ligne courante incluse (si elle existe) jusqu'à la ligne pointée par le paramètre de repérage exclue.

b) - D d

provoque la suppression dans le texte objet, de d lignes du texte source, depuis la ligne courante incluse (si elle existe).

Dans les 2 cas, après l'exécution de la commande, la "ligne pointée" du texte source devient "ligne courante" et est imprimée sur l'unité symbolique EL afin de permettre à l'utilisateur d'en connaître le contenu.



### 3.4 - COMMANDE INSERT

Syntaxe : I (CR)

Cette commande provoque l'insertion dans le texte objet, des lignes lues sur l'unité symbolique PC après recopie de la ligne courante du texte source (si elle existe).

Afin de rappeler à l'utilisateur qu'il se trouve en mode insertion EDITEX génère un caractère ' sur l'unité symbolique EL avant chaque insertion. La fin de l'insertion est indiquée par une ligne vide ( ) (CR)

Exemple : I

( ' ) première ligne à insérer  
( ' ) deuxième ligne à insérer  
( ' ) (CR)

La dernière ligne insérée devient ligne courante et peut être modifiée par commande.

Remarque :

Si après avoir lancé une commande I l'utilisateur n'insère aucune phrase il n'y a plus de ligne courante.

### 3.5 - COMMANDE REPLACE

Syntaxe : R [n] [p] (CR)

a) - R / chaîne1 / [ [+ d ]  
[ o ] ]

provoque le remplacement, dans le texte objet, de la ligne pointée par le "paramètre de repérage de ligne" par une suite de lignes introduites en insertion.

Attention : le paramètre + d est ici géré d'une façon particulière.

Cette commande s'exécute de la façon suivante :

- d'abord recopie des lignes du texte source dans le texte objet, depuis la ligne courante comprise, jusqu'à la ligne contenant la CHAINE1 (non comprise) qui devient la ligne courante temporaire,
- puis recopie jusqu'à la dième ligne non comprise, du texte source, dans le texte objet, depuis la ligne courante temporaire comprise (celle-ci portant le numéro 1), et remplacement de la dième ligne du texte source par celle(s) introduite(s) en insertion.

Remarques :

- Les commandes R/CHAINE1/ et R/CHAINE1/+ 1 ont le même effet.
- La commande R/CHAINE1/+2 va remplacer la ligne qui suit celle contenant CHAINE1.

Se reporter à l'exemple situé au paragraphe 3.11 qui en illustre le fonctionnement.

b) - R d

provoque le remplacement, dans le texte objet de la dième ligne du texte source (la ligne courante porte le n° 1) par une suite de lignes introduites en insertion,

les lignes à insérer sont lues sur l'unité symbolique PC et sont insérées suivant le même processus qu'au § 3.4.

Une fois atteinte, la ligne à remplacer est imprimée sur EL afin de permettre à l'utilisateur de vérifier qu'il s'agit bien de la ligne qu'il désirait.

On notera que cette ligne se trouve alors détruite et en cas d'erreur dans le repérage de la ligne à remplacer, cette erreur ne peut être éventuellement récupérée qu'après avoir inséré la ligne ainsi détruite.

### 3.6 - COMMANDE EXCHANGE

Syntaxe : X [ n ] / chaîne1/ chaîne2 (CR)

Cette commande porte sur la ligne courante du texte source.

Dans cette ligne, la première occurrence de chaîne1 est reconnue suivant le processus habituel (nombre d'espaces d'une suite d'espaces non significatif) et est remplacée par chaîne2 où les espaces sont significatifs en position et en nombre.

Exemple :

modifiée par : X/ 1 ER/ 2 EME  
la ligne courante : < TEST 1 ER PASSAGE  
devient : < TEST 2 EME PASSAGE

La ligne courante une fois modifiée est imprimée sur l'unité symbolique EL mais reste ligne courante et peut à nouveau être modifiée.

Ceci permet à l'utilisateur de vérifier que le résultat qu'il désire a été obtenu et dans le cas contraire de recommencer sa manipulation.

Par convention la chaîne vide (//) est localisée en début de ligne ; aussi la commande X// chaîne (CR) aura pour effet d'insérer chaîne en tête de la ligne courante.

L'absence de paramètre dans la commande X (X (CR)) est équivalent à (X// (CR)) et provoque une simple impression de la ligne courante sur l'unité symbolique EL.

Dans le cas de X n / CHAINE1 / CHAINE2 (CR) seule la zone de tabulation n est modifiée en ajoutant des espaces si nécessaire ou en provoquant un message d'erreur si des caractères autres que des espaces disparaissent lors du remplacement.

### 3.7 - COMMANDES DE TABULATION : TABS ET TABC

On peut utiliser une tabulation de deux façons :

- en insertion de lignes :

La frappe du caractère de tabulation (implicitement % mais qui peut être modifié par la commande TABC) provoque dans la ligne, la génération du nombre d'espaces nécessaire pour se trouver au début de la zone suivante de tabulation.

- pour limiter la portée des commandes dans les lignes traitées :

La portée des commandes C, D, R et X peut être limitée sur une zone de tabulation, en précisant le numéro de celle-ci derrière la commande.

Exemple : C1 / ETIQ /

provoque la recherche de ETIQ dans la première zone de tabulation seulement.

Commande de définition de tabulations : TABS

Syntaxe : TABS d1 [ , d2, d3, . . . . . , d9 ] (CR)

où :  $i$  est le numéro de la première colonne de la zone de tabulation  $i$ .

Les valeurs décimales doivent être données par valeurs croissantes et inférieures à 72 ou 80 (voir paragraphe 4.1). De plus, seules 9 zones de tabulations peuvent être définies par la commande TABS.

Nb : Une tabulation spéciale est prédéfinie implicitement. Elle correspond à l'édition de programmes écrits en assembleur T1600 : 1, 8, 15, 25.

- 1 : début de la zone étiquette
- 8 : début de la zone opération
- 15 : début de la zone opérande
- 25 : début de la zone commentaire

Commande de définition du caractère de tabulation : TABC

Syntaxe : TABC C (CR)

Cette commande permet de remplacer le caractère de tabulation (%) implicitement défini par le caractère C qui doit être différent d'un espace ( ), (←), (↑) et (CR).

### 3.8 - COMMANDE BLANK

Syntaxe : B (CR)

Elle permet de perforer une avance bande sur le périphérique perforateur du ruban de papier.

### 3.9 - COMMANDE SEPARATOR

Syntaxe : S c (CR)

Cette commande permet de remplacer le séparateur de chaîne (implicitement le caractère / slash) par le caractère c qui doit être différent d'un espace ( ), (←), (↑) et (CR).

### 3.10 - COMMANDE EOT

Syntaxe : E (CR) .

Après avoir recopié la ligne courante (si elle existe) cette commande génère dans le texte objet l'impression du caractère (EOT) indiquant la fin de l'édition puis rend le contrôle au système.

Remarque :

En cours d'exécution d'une commande, la lecture du caractère (EOT) dans le texte source provoque l'impression du message :

ERT 08

La commande est interrompue, le caractère (EOT) n'est pas recopié et l'éditeur repasse en attente de commande.

### 3.11 - EXEMPLE D'UTILISATION DES COMMANDES

Texte source

```

DÉBUT : LAI      1      < DÉBUT DE L'EXEMPLE
        STA      TOTAL
BOUCLE : LA      MEM1   <   CETTE
        AD      MEM2   <   BOUCLE
        STA      MEM1   <   VA
        LA      TOTAL   <   ETRE
        ADRI    1,A    <   EXÉCUTÉE
        STA      TOTAL   <   PLUSIEURS
        CPI     100    <   FOIS
        JE      SUITE  <   DE
        JMP     BOUCLE <   SUITE
SUITE :  LA      MEM1
        CPI     100
FIN :    JNE     BIZARE < FIN DE L'EXEMPLE
        -----
    
```

Commandes frappées par l'utilisateur et réponses (soulignées) d'EDITEX émises sur EL

```

C/TOTAL/
      STA      TOTAL
      |
      |-%STA%MEM1
      |-%STA%MEM2
      |
C1/BOUCLE/ + 2
      STA      MEM1      <   VA
      |
      |-%IC%TOTAL% < DEVOIR
      |
C 2
      ADRI     1,A      <   EXÉCUTÉE
      |
D/100/
      CPI      100     <   FOIS
      |
X4/FOIS/EXÉCUTÉE
      CPI      100     <   EXÉCUTÉE
      |
C
      JE      SUITE   <   DE
      |
    
```

```

D
      JMP          BOUCLE      < SUITE
X2/MP/NE
      JNE          BOUCLE      < SUITE
X4/SUITE/100 FOIS
      JNE          BOUCLE      < 100 FOIS

R/FIN DE/+ 1
FIN : JNE BIZARE < FIN DE L'EXEMPLE
FIN : % JNE % BIZARE % < ATTENTION A L'EXEMPLE
.
E
  
```

Texte **objet**

```

DÉBUT : LAI      1      < DÉBUT DE L'EXEMPLE
        STA      TOTAL
        STA      MEM1
        STA      MEM2
BOUCLE : LA      MEM1   <   CETTE
        AD      MEM2   <   BOUCLE
        STA      MEM1   <   VA
        IC      TOTAL  <   DEVOIR
        LA      TOTAL  <   ETRE
        CPI      100   <   EXÉCUTÉE
        JNE      BOUCLE <   100 FOIS
SUITE : LA      MEM1
        CPI      100
FIN :  JNE      BIZARE <  ATTENTION A L'EXEMPLE
  
```



## 4.2 - MESSAGES ÉMIS PAR ÉDITEX

Lorsque EDITEX reçoit une commande dont la syntaxe est erronée ou dont l'exécution n'est possible qu'en engendrant une erreur, il envoie un message à l'utilisateur (sur l'unité symbolique EL) pour l'en informer, après quoi il annule la commande en cours.

De plus, en fin d'exécution d'une commande, EDITEX transmet sur l'unité symbolique EL des messages de dialogue ayant pour but d'informer l'utilisateur sur le déroulement de son travail.

Les messages de dialogue sont en général constitués par l'impression de la ligne courante du texte source à la fin de l'exécution de la commande en cours.

Les messages d'erreur donnent par un numéro, le type de l'erreur détectée (ERT nn).

## 4.3 - CORRECTION DES ERREURS

EDITEX détecte les erreurs suivantes :

- erreurs de parité
- erreurs de syntaxe dans les commandes
- erreurs liées à l'exécution des commandes

### 4.3.1 - LES ERREURS DE PARITÉ

Lorsque une telle erreur est détectée, un message est généré sur l'unité symbolique EL, la phrase contenant cette erreur est imprimée sur EL avec le caractère erroné remplacé par un ( \ ) ; puis EDITEX se met en attente d'une nouvelle commande sur l'unité symbolique EC (Error Correction).

Remarque :

Lorsqu'une erreur de parité est détectée dans une phrase à insérer (commandes I et R) après l'émission du message d'erreur, la phrase erronée est imprimée sur EL puis détruite EDITEX repasse alors en mode "insertion" (impression du caractère ( ) sur EL) et attend, sur l'unité symbolique PC, une nouvelle phrase à insérer.

### 4.3.2 - ERREURS DE SYNTAXE DANS LES COMMANDES

Lorsqu'une erreur de syntaxe est décelée dans une commande, un message d'erreur est généré sur l'unité symbolique EL (la liste des numéros d'erreurs est donnée en annexe) puis la commande en cours est annulée : EDITEX en attend une nouvelle sur l'unité symbolique EC (Error Correction).

### 4.3.3 - ERREURS LIÉES A L'EXÉCUTION DES COMMANDES

Lorsqu'une telle erreur est détectée un message d'erreur est généré sur EL, l'exécution de la commande en cours est stoppée et EDITEX en attend une nouvelle sur l'unité symbolique EC.

## 4.4 - INTERRUPTION D'UNE ÉDITION EN COURS

L'utilisateur dispose sur le périphérique de dialogue du bouton "BREAK" qu'il peut utiliser pour interrompre une édition en cours (par exemple pour changer le support du symbolique).

Dans ce cas EDITEX rend le contrôle au système.

Remarque :

Il pourra d'ailleurs être réactivé ultérieurement par la commande CTEX car le test des appels opérateurs est fait en des points où EDITEX peut être interrompu.

## ANNEXE

### A - Synoptique des commandes

Rappel :

n : numéro de la zone de tabulation associée éventuellement à la commande

p : paramètre de repérage de ligne

avec  $s = \text{/chaîne/}$

$$p = \left[ \begin{array}{c} s \\ \left[ \begin{array}{c} + d \\ + 0 \end{array} \right] \end{array} \right]$$

Commandes	Syntaxes	Fonctions
COPY	$C [n] [p] \text{ (CR)}$	<ul style="list-style-type: none"> <li>- C s recopie dans le texte objet, les lignes du texte source comprises entre la ligne courante incluse ( si elle existe) et la ligne pointée par s exclue ; celle-ci devient ligne courante.</li> <li>- C d recopie dans le texte objet, d lignes du texte source à partir de la ligne courante incluse ( si elle existe ).</li> </ul>
DELETE	$D [n] [p] \text{ (CR)}$	<ul style="list-style-type: none"> <li>- D s supprime dans le texte objet, les lignes du texte source comprises entre la ligne courante incluse (si elle existe) et la ligne pointée par s exclue, celle-ci devient ligne courante.</li> <li>- D d supprime dans le texte objet, d lignes du texte source à partir de la ligne courante incluse (si elle existe).</li> </ul>
INSERT	$I \text{ (CR)}$	insert dans le texte objet les lignes lues sur l'unité symbolique PC après recopie de la ligne courante (si elle existe). La dernière ligne insérée devient ligne courante du texte source.
REPLACE	$R [n] [p] \text{ (CR)}$	<ul style="list-style-type: none"> <li>- R s remplace dans le texte objet, la ligne du texte source pointée par s, par une suite de lignes introduites en insertion sur l'unité symbolique PC. La dernière ligne insérée devient ligne courante du texte source.</li> <li>- R d remplace dans le texte objet, la d ième ligne du texte source depuis la ligne courante incluse (qui porte le n° 1) par une suite de lignes introduites en insertion sur l'unité symbolique PC. La dernière ligne insérée devient ligne courante du texte source.</li> </ul>



EXCHANGE	$X [n] [ \text{chaîne1} / \text{chaîne2} ]$	(CR)	remplace dans la ligne courante du texte source la chaîne de caractères Chaîne1 par Chaîne2. Dans Chaîne1 le nombre d'espaces d'une suite d'espaces n'est pas significatif ; dans Chaîne2 au contraire les espaces sont significatifs en position et en nombre.
TABS	TABS d1 [ , d2, d3, ...; dg ]	(CR)	redéfinition des zones de tabulation.
TABC	TABC c	(CR)	remplace le caractère de tabulation précédemment défini par le caractère c qui doit être imprimable et différent d'un espace ( ), (←), (↑) et (CR).
BLANK	B	(CR)	perfore une "avance-bande" sur l'unité symbolique SØ lorsque l'unité fonctionnelle qui lui est affectée est le perforateur de ruban.
SEPARATOR	S c	(CR)	remplace le séparateur de chaîne précédemment défini par le caractère c qui doit être imprimable et différent d'un espace ( ), (↑), (←) et (CR).
EOT	E	(CR)	recopie la ligne courante (si elle existe), imprime le caractère (EOT) dans le texte objet, perfore une avance-bande sur l'unité symbolique SØ lorsque l'unité fonctionnelle qui lui est affectée est le perforateur de ruban puis rend le contrôle au système.

**B - Liste des numéros d'erreurs**

- |     |    |  |
|-----|----|--|
| ERT | 00 | Erreur de parité   |
|     | 01 | Type de commande non reconnu   |
|     | 02 | Tabulation incorrecte  |
|     | 03 | Paramètre de repérage de ligne incorrect   |
|     | 04 | Syntaxe de la commande incorrecte  |
|     | 05 | Chaîne de caractères trop longue   |
|     | 06 | Caractères non blancs écrasés lors de l'exécution de la commande X limitée par une tabulation. |
|     | 07 | Chaîne à remplacer absente de la ligne courante du texte source à éditer (commande X).         |
|     | 08 | Fin de fichier.  |