

PS/DP/NOTE 88-17/REVISED  
NOVEMBRE 1988

**NODAL**

au PS

**Mode d'Emploi**

Version 1.0

par R.Hoh

**AVERTISSEMENT**

=====

Ce manuel a une histoire: débutant moi-même en NODAL, j'avais besoin d'un mode d'emploi pour suppléer une mémoire par trop volatile: la mienne. Ce document n'existant pas, il suffisait de le créer. M'étant rapidement rendu compte qu'il pourrait être utile à d'autres, je l'ai adapté à l'utilisation pour une personne novice, n'ayant tout au plus que quelques notions de BASIC.

Ce document est, par conséquent, destiné aux débutants et/ou à ceux qui, comme moi, ont la syntaxe par trop aléatoire.

Que les autres destinataires me renvoient leur exemplaire sans plus tarder: il s'agit d'une erreur dans la distribution!!!

L'auteur.

```
*****
* IMPORTANT: Terminologie utilisée dans ce manuel. *
*****
```

Lorsqu'un élément apparaît entre parenthèses angulaires, comme:

<numéro de bloc>

...il ne faut ni taper ces parenthèses, ni ce qu'elles renferment. L'exemple ci-dessus indique le type de données requises pour la commande. Par exemple:

EDIT <numéro de bloc>.<numéro de ligne>

...signifie que vous devez saisir (derrière le ">" de NODAL), par exemple :

EDIT 12.32

Les parenthèses () doivent être tapées telles quelles. Dans l'exemple:

COS(<expression numérique>)

...l'expression numérique doit se trouver entre parenthèses. Par exemple:

TYPE COS(PIE/4)

Les crochets rectangulaires contiennent les éléments facultatifs d'une commande ou d'une fonction. Par exemple:

DO <numéro de bloc>[.<numéro de ligne>]

...signifie que vous n'êtes pas obligé de faire suivre le mot clé DO des deux paramètres, mais que vous pouvez, si vous le désirez, ajouter le paramètre <numéro de ligne>. La commande peut alors être entrée de deux manières différentes, par exemple:

DO 12 ...ou... DO 12.34

### \*\*\* CARACTERES SPECIAUX \*\*\*

```
! Passer à la ligne suivante (Carriage Return + Line Feed)
%n Changement du format d'affichage (par défaut: %11.04)
%, Format d'affichage en notation E (longueur du champ: 17)
\n Equivalent ASCII de n (TYPE \65 donne A)
&n Afficher n espaces (code ASCII 32)
]n Pour sortir n en Octal
]]n Pour sortir n en Hexadécimal
?n Pour sortir n en Binaire
[n Pour entrer n en Octal
[[n Pour entrer n en Hexadécimal
```

Ces caractères sont utilisables dans toute chaîne de caractères.

Un format %8.03 signifie 3 chiffres après le point, sur un champ total de 8 caractères. Un format %6 permet l'affichage des entiers signés.

L'indication %,1 affichera 1.0000000E0 . Il ne faut pas confondre % à l'intérieur d'une commande d'affichage avec la commande % qui signifie "commentaire".

Un nombre trop grand pour être écrit dans le format spécifié est automatiquement affiché en format E. L'indication de format reste vraie sur toute la ligne de commande, sauf si un changement de format est spécifié. L'indication de format prend par défaut la valeur %11.04 en début de commande et en l'absence de spécification contraire.

### \*\*\* TYPE DE DONNEES \*\*\*

La longueur d'une chaîne de caractères, comme la longueur d'une ligne NODAL, ne peut dépasser 80.

NODAL ne calcule qu'avec des nombres réels, mais ces derniers sont temporairement convertis en entiers lors d'opérations logiques ou lors d'affectation à un tableau d'entiers. Les entiers peuvent varier de -32768 à +32767.

Une <expression numérique> est une expression qui aboutit à une valeur numérique. Cela peut être des nombres, ou une variable numérique, ou une/des opération(s) s'effectuant avec des nombres et/ou des variables numériques.

Une <liste de: élément> décrit un paramètre comprenant une liste d'un ou de plusieurs éléments séparés par des espaces ou par des virgules (voir l'exemple). Cette liste est limitée par la longueur de la ligne.

Les "variables programme" comprennent au maximum 6 caractères, mais au-delà de 2 caractères, un point est obligatoire.

Exemples: ABCD.E    SIGN.I

Les "variables système" comprennent un minimum de 3 caractères et un maximum de 6, sans point de séparation.

Exemples: ABCDEF    INJ:VB

Les "variables indicées" (tableau) comprennent au maximum 6 caractères, avec ou sans point de séparation.

Les fonctions définies par l'utilisateur (defined functions) comprennent également un maximum de 6 caractères.

**\*\*\* A PROPOS DES MOTS CLES \*\*\***

Les mots clés sont classés par ordre alphabétique suivis par \$, % et ?.

Ils sont répertoriés de la manière suivante:

- 1°) Mot Clé en grand, permettant un repérage plus facile
- 2°) Ligne(s) d'explication(s) avec l'abréviation minimale du Mot Clé
- 3°) Exemple(s) d'utilisation(s) du Mot Clé
- 4°) Notes explicatives détaillées

**\*\*\* EXCEPTIONS \*\*\***

Certaines fonctions (telles certaines "Pattern Functions" par exemple) ne figurent pas dans ce manuel. En effet, l'expérience montre que ces fonctions sont très rarement employées.

Les fonctions propres à l'utilisation de l'écran graphique ne sont pas expliquées dans ce manuel. Voyez à ce sujet la note "Spécification du package graphique des consoles principales" (PS-CO-Note 79-14 du 29/6/79).

\*\*\*\*\*  
 \* MOTS CLES... \*  
 \*\*\*\*\*

## ABS

ABS(<expression numérique>)

>TYPE ABS(-67.98)  
 67.98

FUNCTION: Donne la valeur ABSolue de l'expression entre parenthèses. Les nombres négatifs perdent donc leur signe.

Mots clés associés: FPT, INT, SGN

## AEQM

AEQM(<numéro d'équipement module>, <valeur tableau>, <flag>, <tableau du numéro d'équipement>, <propriétés>, <ligne PLS>, <code d'erreur du tableau>)

>AEQM(10, AR, FA, EAR, PPM, PS, CO)

FUNCTION: Permet d'appeler un module d'équipement par son numéro (Array Call). Voir COOKBOOK 4.4. PS/CO 08-84.

Mot clé associé: EQM

## AND

AND(<argument entier>, <argument entier>)

>SET C=AND(A, B)

FUNCTION: Effectue bit par bit l'opération booléenne ET entre les deux <arguments entiers>.

Mots clés associés: IOR, NEG

## ARG

SET ARG(<indice>)=<expression numérique> ou  
 SET <variable>=ARG(<indice>)

>SET ARG(16)=9\*9  
 >SET AB.CDE=ARG(15)

FUNCTION: Cette fonction permet l'écriture ou la lecture d'un tableau résident de 16 variables globales réelles, indicées de 1 à 16. Ceci permet le passage de variables d'une tâche à une autre. Elles ne sont pas affectées par ERASE ALL, OLD ou LOAD.

Mots clés associés: STRARG, OVERLAY, START

**ARSIZE**

ARSIZE(<nom du tableau>)

>DIM-I ABC(9\*N)

>SET N.MAX=ARSIZE(ABC)

>DIM-STR ABC

>\$SET ABC(9)="Marcel";\$SET ABC(29)="Jules"

>TYPE ARSIZE(ABC)

2

**FONCTION:** Permet de connaître le nombre d'éléments d'un tableau. Pour les tableaux numériques, ARSIZE retourne la valeur entrée lors de sa déclaration avec la commande DIM. Les tableaux de chaînes de caractères étant dynamiques, seuls les éléments ayant été affectés sont pris en compte et non l'indice maximal. Voir le deuxième exemple ci-dessus.

**Mot clé associé:** DIMENSION

**ASCII**

ASCII(<chaîne de caractères>)

>TYPE ASCII(SUBS(1,1,"Albert"))

65

**FONCTION:** Donne le total des codes ASCII de tous les caractères formant la <chaîne de caractères>.

**Mot clé associé:** Aucun

**ASK**

A [<chaîne>]<variable>[<chaîne>]<variable> etc...

>10.50 ASK X Y Z(1)

>10.85 ASK"Value of x"\*X"Value of y"\*Y"Value of z(1)\*Z(1)

**COMMANDE:** Quand NODAL rencontre une ligne comprenant une commande ASK, la chaîne de caractères est affichée (si elle existe). Un double point est affiché. L'utilisateur entre alors une valeur numérique suivie de [RETURN]. Un passage à la ligne suivante a donc lieu à chaque entrée.

**Mot clé associé:** \$ASK

**AT2**

AT2(<expression numérique 1>,<expression numérique 2>)

>TYPE AT2(1,2)  
.4636

FONCTION: Calcule l'Arc Tangente de l'<expression numérique 1> divisé par l'<expression numérique 2> en Radians, dans les limites de -PIE à PIE.

Mots clés associés: COS, SIN, PIE

**BACKGR**

BACKGR<couleur de fond>

>WRITE(COLOUR)RED BACKGR YELLOW "Hello world!"

VARIABLE CHAINE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur de fond des caractères suivants. Doit impérativement être suivi par une couleur. En l'absence de cette spécification, la couleur de fond est le noir.

Mots clés associés: BLACK, BLUE, ERASE, GREEN, INVERT, LBLUE, MAGENT, RED, WHITE, YELLOW

**BALLGR**

BALLGR

>61.20 SE GR=0  
>61.30 SE BT=MWAIT(GR);IF GR<>BALLGR;SE XI=1;RETURN

FONCTION: Cette fonction est en fait une variable système, qui retourne la valeur du groupe d'événements associé au bouton de validation de la boule.

Mots clés associés: KNBGR, TPGR, TRIGER

**BALLST**

BALLST

>SE D.LUN=BALLST

FONCTION: Donne le numéro d'unité logique de l'équipement sur lequel le curseur est actuellement connecté. Si aucun équipement n'a de curseur connecté, la fonction retourne zéro, ce qui signifie que le curseur est libre, et peut donc être affecté à un équipement.

Mots clés associés: KDSABL, KENABL

**BANDW**

BANDW(<numéro d'écran noir et blanc, de 1 à 4>)

```
>SE D.LUN=BANDW(4)
>WRITE(BANDW(1))"Hello World!"
```

FONCTION: Donne le numéro d'unité logique qui identifie un des quatre écrans noir et blanc. Le deuxième exemple est d'une utilisation fréquente.

Mots clés associés: COLOUR, GRAPH, LEGLUN

**BIT**

BIT(<0 à 15>,<argument entier>)

```
>SE IM.PAR=BIT(0,AB.CDE);% Si le bit 0 vaut 1, AB.CDE est impair
>SE BIT(0,AB.CDE)=0;% Force le bit 0 à 0
```

FONCTION: Cette fonction est utilisable aussi bien en lecture qu'en écriture. Dans le premier cas, elle donne la valeur du bit 0-15 de l'argument entier. Dans le second, elle force un bit de l'argument entier.

Mot clé associé: Aucun

**BLACK**

BLACK

```
>WRITE(COLOUR)POSIT(26,11)BLACK BACKGR WHITE"Hello world!"
```

VARIABLE CHAINE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur d'écriture (ou de fond si précédé de "BACKGR") des caractères suivants. Dans le cas particulier "BLACK", il faut spécifier une couleur de fond après la couleur de caractère, avec la fonction BACKGR. En effet, le fait de définir une couleur de caractère définit automatiquement la couleur de fond comme noire. Une spécification de couleur de fond doit toujours suivre, et non précéder, la spécification de couleur d'écriture.

Mots clés associés: BACKGR, BLUE, ERASE, GREEN, INVERT, LBLUE, MAGENT, RED, WHITE, YELLOW

**BLUE**

BLUE

```
>WRITE(COLOUR)POSIT(26,11)BLUE"Hello World!"
```

VARIABLE CHAINE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur d'écriture (ou de fond si précédé de "BACKGR") des caractères suivants. Le fait de spécifier une couleur d'écriture définit automatiquement la couleur de fond comme noire. Une spécification de couleur de fond doit donc toujours suivre, et non précéder, la spécification de couleur d'écriture.

Mots clés associés: BACKGR, BLACK, ERASE, GREEN, INVERT, LBLUE, MAGENT, RED, WHITE, YELLOW

**BUTS**

BUTS

```
>SET BT=BUTS(GR)
```

FONCTION: Permet la lecture immédiate du dernier bouton activé ainsi que du groupe (GR), puis nettoie le "tampon-bouton". Peut être utilisé uniquement pour cette deuxième caractéristique.

Mot clé associé: Aucun

**CALL**

```
C <fonction>([<liste de: paramètre>])
```

```
>CALL CALCUL(A,B,C,D,E,F)
```

COMMANDE: Permet l'appel de fonctions de type CALL, paramétrées ou non. Il n'est pas nécessaire que le mot clé CALL précède le nom de la fonction, mais dans ce cas, il y a une pénalité en temps d'exécution.

Mot clé associé: Aucun

**CLOSE**

```
CLOSE(<numéro logique de fichier>)
```

```
>CLOSE(-1)
```

FONCTION: Ferme le fichier dont le numéro logique de l'unité (LUN) est indiqué entre parenthèses. Ce numéro est obtenu lors de l'ouverture du fichier, avec la commande OPEN. La valeur -1 ferme tous les fichiers ouverts.

Mots clés associés: IDEV, ODEV, OPEN

**CLZON**

```
CLZON(<numéro de l'écran ou de la partie d'écran>,n,n,n,n,CO)
>FOR I=1,4;CLZON(I,0,0,0,0,CO))
```

FONCTION: Cette fonction permet de réserver un écran et d'annuler une éventuelle réservation faite par un autre utilisateur. L'exemple ci-dessus est le plus couramment employé. Pour de plus amples renseignements (sur les valeurs "n"), voir "Introduction to Application Programme Writing" PS/CO/NOTE 86-001.

Mot clé associé: Aucun

**COLOUR**

```
COLOUR
```

```
>WRITE(COLOUR)ERASE
>SE TV=COLOUR
>WRITE(TV)POSIT(26,11)"Hello world!"
```

FONCTION: Cette fonction est en fait une variable. Permet de définir que la fonction WRITE est destinée à l'écran couleur. La deuxième ligne permet de gagner quatre caractères, en remplaçant "COLOUR" par "TV".

Mots clés associés: BANDW, GRAPH, LEGLUN

**COPY**

```
COPY(<tableau S>,<tableau O>,<pointeur S>,<pointeur O>)
```

```
>1.10 DIM A(5);DIM B(5)
>1.20 FOR I=1,5;SE A(I)=I;SE B(I)=I+100
>1.30 COPY(A,B,1,2)
>1.40 FOR I=1,5;TY A(I) B(I)!
>RU
```

1	101
2	1
3	2
4	3
5	4

FONCTION: Copie le contenu du <tableau Source> (à partir du <pointeur Source>) dans le <tableau Objet> (à partir du <pointeur Objet>), jusqu'à la première fin de tableau rencontrée. Cette fonction ne copie que des tableaux de valeurs numériques.

Mot clé associé: Aucun

**COS**

COS(<expression numérique>)

```
>TYPE COS(PIE/4)
      .7071
```

FONCTION: Calcule le COSinus de l'expression numérique. L'expression numérique doit être exprimée en radians.

Mots clés associés: AT2, SIN, PIE

**CREATE**

CREATE(<nom de la variable globale>,<type>,CO)

```
>CREATE("ABC",REAL,CO)
```

FONCTION: Cette fonction crée une variable globale de type:

- BLOCK c'est-à-dire un tableau de 64 entiers
- REAL c'est-à-dire une valeur réelle
- STRING c'est-à-dire une chaîne de caractères

Mots clés associés: DELETE, FETCH, LISG, STORE

**DATE**

DATE

```
>TYPE DATE
1987-12-31-23:59:59
```

FONCTION: Retourne une chaîne de caractères contenant la date en format ISO, c.à.d. l'année, le mois, le jour, l'heure, les minutes et les secondes. Dans l'exemple ci-dessus est représenté le dernier instant avant 1988.

Mot clé associé: TIME

**DEFINE-CALL**

DE-C <nom de la fonction>(<liste de: paramètres>)

```
>DEFINE-CALL COMP(V-ABC, R-DEF, S-GHI)
```

COMMANDE: Le nom des paramètres formels dans l'exemple ci-dessus sont ABC, DEF et GHI. Les préfixes "V-", "R-", et "S-" indiquent le type de paramètre formel soit:

- "Value": Valeur ou variable, seulement lisible par la fonction.
- "Reference": Variable, valable en lecture et/ou en écriture.
- "String": Variable chaîne ou chaîne de caractères.

Jusqu'à huit paramètres formels peuvent être passés à la fonction. pour de plus amples informations, voir "THE NODAL SYSTEM FOR THE SPS" CERN SPS 78-07.

Mots clés associés: DEFINE-FUNCTION, DEFINE-STRING

**DEFINE-FUNCTION**

```
DE-F <nom de la fonction>

>1.1 EXECUTE(5)1.2;WAIT(5);VALUE X
>1.2 SET X=BCT(3);REMIT X
>DEFINE-FUNCTION BCTR3
```

COMMANDE: La commande immédiate ci-dessus crée la fonction définie NODAL BCTR3. Le corps de la fonction est le texte présent dans le tampon au moment où la commande DEFINE-FUNCTION est donnée. La commande VALUE termine l'exécution de la fonction. Cette commande ne retourne qu'une seule variable numérique. Pour de plus amples informations, voir "THE NODAL SYSTEM FOR THE SPS" CERN SPS 78-07.

Mots clés associés: DEFINE-CALL, DEFINE-STRING

**DEFINE-STRING**

```
DE-S <nom de la fonction>

>1.1 $VALUE"Ceci est : "A
>DEF-STR RESULT(S-A)
-----
>$SE A="vrai";TYPE RESULT(A)
Ceci est : vrai
```

COMMANDE: La commande immédiate ci-dessus crée la "fonction chaîne définie" RESULT. Cet exemple ne fait qu'une concaténation. Pour de plus amples informations, voir "THE NODAL SYSTEM FOR THE SPS" CERN SPS 78-07.

Mots clés associés: DEFINE-CALL, DEFINE-FUNCTION

**DELETE**

```
DELETE(<nom de la variable globale>,CO)

>DELETE("ABC",CO)
```

FONCTION: Cette fonction efface une variable globale.

Mots clés associés: CREATE, FETCH, LISG, STORE

**DIMENSION**

DI <nom du tableau de réels>(<liste de: expression numérique>)  
 DI-I <nom du tableau d'entiers>(<liste de: expression numérique>)  
 DI-STR <nom du tableau de chaînes de caractères>

>DIM B(10,2)  
 >DIM-I J(4,5)  
 >DIM-STR A

COMMANDE: Les deux premières commandes allouent l'espace requis aux tableaux de réels et d'entiers et spécifient les valeurs d'indices maximales. La troisième commande déclare un tableau de chaînes de caractères qui est dynamique et ne peut avoir qu'une dimension. La valeur minimale d'un indice est un. (C'est le premier élément du tableau). Le fait de dimensionner un tableau existant efface l'ancien tableau de même nom. Le fait de déclarer un tableau numérique met toutes les valeurs de ce tableau à zéro.

Mots clés associés: ARSIZE, COPY, ERASE

**DO**

DO <numéro de bloc>[.<numéro de ligne>][!<voir texte>]  
 DO <expression numérique>[!<voir texte>]

>DO 10 !10 !12  
 >DO 24!  
 >DO 12.34  
 >DO 11+(1/100)

COMMANDE: Exécute un groupe ou une ligne. Dans le premier exemple, le groupe 10 est exécuté. S'il y a faute dans ce groupe, alors il est réexécuté, s'il y a toujours faute, alors le groupe 12 est exécuté. Dans le deuxième exemple, le bloc 24 est exécuté jusqu'à la faute, sans s'arrêter sur celle-ci. Les exemples suivants montrent deux possibilités d'exécution d'une ligne. Ne sortez jamais d'un bloc appelé par DO avec la commande GOTO.

Mot clé associé: RETURN

**EDIT**

ED <numéro de bloc>.<numéro de ligne>[,L]

>ED 10.5  
 >ED 10.5,L

COMMANDE: Permet d'accéder à l'éditeur de NODAL. Le deuxième exemple "liste" la ligne avant d'entrer dans l'éditeur. Une ligne entrée avec les mêmes numéros de bloc et de ligne, écrase l'ancienne ligne. Voir les "Caractères de Contrôle utilisables en Mode EDITion", à la fin de ce manuel.

Mot clé associé: ERASE

**END**

EN

&gt;1.99 END

COMMANDE: Un programme s'arrête normalement lorsque le plus grand numéro de ligne du plus grand numéro de bloc est atteint. Cette commande permet d'arrêter le programme plus tôt. Afin de faciliter la maintenance des programmes, le premier bloc ne comporte, en général, que des commandes DO, la dernière suivie d'une commande END. Ceci rend également le programme plus lisible. Pour réaliser un "bon" programme, ces conditions sont nécessaires, mais non suffisantes...

Mot clé associé: Aucun

**EMMESS**

EMMESS(&lt;numéro de l'erreur&gt;)

&gt;TYPE EMMESS(3)

ILLEGAL ARITHMETIC EXPRESSION

FONCTION: Permet l'interprétation d'une erreur liée à l'utilisation d'un module d'équipement. Cette fonction n'est pas implémentée dans l'ordinateur PRDEV.

Mot clé associé: ERMES

**ENDPG**

ENDPG(&lt;nom du programme&gt;,CO)

&gt;1.99 ENDPG(PROG,CO);END

FONCTION: Cette fonction est à utiliser lorsqu'un programme a été activé par une touche du "TP TREES" et doit être rayé de la liste des programmes actifs. L'inversion de la touche d'activation est également effectuée.

Mot clé associé: Aucun

**EQM**

EQM(<numéro du module d'équipement>,<valeur>,<flag>,<numéro d'équipement>,<propriété>,<ligne PLS>,<code d'erreur>)

>EQM(10,B,FL,33,PPM,SFT,CO)

FONCTION: Permet d'appeler un module d'équipement par son numéro. Voir COOKBOOK 4.4 PS/CO 08-84.

Mot clé associé: AEQM

**ERASE**

ER <liste de: <numéro de bloc>[.<numéro de ligne>]>

ER <liste de: nom de variable>

>ERA ALLP

>ERA ALLV

>ERA ALL

COMMANDE: Efface les blocs, lignes, variables et/ou tableaux indiqués. La commande ERA ALLP efface tout le programme, mais pas les variables. La commande ERA ALLV efface toutes les variables, mais laisse le programme intact. Le dernier exemple efface tout et devrait être utilisé avant l'écriture de tout nouveau programme. Un effacement sélectif de tableaux permet de récupérer de la place dans la zone de travail, ce qui peut être utile si celle-ci est comptée. Voir aussi la variable chaîne ERASE ci-dessous.

Mots clés associés: DIM, DI-I, DI-STR

**ERASE**

ERASE

>WRITE(COLOUR)ERASE

VARIABLE CHAINE: Permet l'effacement complet d'un écran, mais non sa réservation. Pour cela, voir CLZON.

Mots clés associés: CLZON, INVERT

**ERMES**

ERMES(<numéro d'erreur>)

>TYPE ERMES(37)  
Value OUT of RANGE

FONCTION: Donne le message d'erreur lié au <numéro d'erreur> indiqué.

Mot clé associé: EMESS, ERROR

**ERROR**

ERROR=(<expression numérique>)

>SET ERROR=(-2)  
Value OUT of RANGE

FONCTION: Provoque une erreur traitée par NODAL. l'<expression numérique> doit être positive. Dans le cas contraire, l'erreur 37 (Value OUT of RANGE) est lancée, car ERROR vaut alors 37!

Mot clé associé: ERMES

**EVAL**

EVAL(<chaîne de caractères>)

>TYPE EVAL("01E2")  
100

FONCTION: Cette fonction fait la conversion de la <chaîne de caractères> en une expression numérique. Si la conversion n'est pas possible, l'erreur 8 (NON existent NAME) est affichée. A noter que \$SET est la fonction inverse de celle-ci.

Mot clé associé: \$SET

**EWAIT**

EWAIT(<nom de la machine>, <impulsion>, <ligne PLS>, CO)

>10.20 EWAIT(PLS, SSC, 35, CO)

FONCTION: Permet d'attendre un moment déterminé du cycle machine. Cette fonction, utilisable uniquement sur un ordinateur FEC, est équivalente à MWAIT pour un ordinateur console. CO est le code d'achèvement (completion code).

Mot clé associé: MWAIT

**EXECUTE**

EX(<ordinateur>)<n<sup>0</sup> de bloc>[<n<sup>0</sup> de ligne>][<liste de: variable et/ou tableau à transférer>]

```
>8.30 EX(CPS) 9 A CO;WAIT(CPS)
```

```
-----
>9.10 SE A=TIME;REMIT A CO
```

COMMANDE: Ceci est la principale commande pour programmer des interactions. L'exemple ci-dessus permet de lire le temps interne dans l'ordinateur CPS, en y faisant exécuter le bloc 9. Si l'ordinateur appelé ne répond pas, cette commande est abandonnée au bout de 3 secondes.

Mots clés associés: IMEX, REMIT, WAIT

**EXIST**

EXIST(<nom du fichier>)

```
>SE RE.P=EXIST("ABC")
```

```
46
```

FONCTION: Cette fonction permet d'obtenir des informations sur un fichier. Si elle retourne:

0 : Accès lecture/écriture possibles

46 : Fichier inconnu

69 : Fichier ouvert par un autre utilisateur

75 : Le fichier existe, mais accès lecture/écriture interdits

etc...

Mot clé associé: Aucun

**EXP**

EXP(<expression numérique>)

```
>TYPE EXP(6.876)
```

```
968.744
```

FONCTION: Calcule "e" à la puissance donnée par l'<expression numérique> où "e" est égal à 2.7182818 environ, le nombre dont le logarithme naturel est 1.

Mot clé associé: LOG

**FCOPY**

FCOPY(<nom du fichier source>,<nom de fichier cible>)

>FCOPY("ABC","DEF")

FONCTION: Cette fonction effectue une copie du fichier source dans un fichier cible, à condition que les deux fichiers existent dans le même ordinateur. Ne vous trompez pas dans l'ordre de ces deux fichiers...

Mot clé associé: Aucun

**FETCH**

FETCH(<nom de la variable>,<nom de la variable globale>,CO)

>FETCH(A,"ABC",CO)

FONCTION: Cette fonction va chercher la valeur de la variable globale et l'affecte à la variable A.

Mots clés associés: CREATE, DELETE, LISG, STORE

**FIRST**

FIRST(<nom du tableau> <opération logique> <expression>)

```
>DIM AR(9);SE B=5
>FOR I=1,9;SE AR(I)=I*3
>SE Z=FIRST(AR>B)
>TY Z
    2
```

FONCTION: Cette fonction donne le premier indice pour lequel l'opération logique est vraie.

Mot clé associé: Aucun

**FOR**

F <variable simple>=<début>,[<incrément>,<fin>]

```
>FOR I=3,-1,1;TYPE %2 I
    3 2 1
```

COMMANDE: Exécute le restant de la ligne autant de fois que l'on peut ajouter l'incrément à la variable simple en partant du début jusqu'à la fin. Si l'incrément est omis, il prend implicitement la valeur 1. La boucle peut être arrêtée avant la fin avec la commande ROF. Attention aux variables réelles: sur certains ordinateurs, 2+3 n'est pas égal à 8, mais à 7.999999. Ceci fait que la boucle n'est parcourue que jusqu'à 7. La boucle FOR est toujours parcourue au moins une fois.

Mot clé associé: ROF

**FPT**

FPT(<expression numérique>)

>TYPE FPT(-6.28)  
-.28

FONCTION: Enlève la partie entière de l'<expression numérique>, sans en changer le signe.

Mots clés associés: ABS, INT, SGN

**FROMC**

FROMC

>SE CN=FROMC

FONCTION: Si cette fonction est appelée lors d'une exécution dans un autre ordinateur (avec EXECUTE, par exemple), elle retourne le numéro de l'ordinateur appelant. Dans le cas où elle est utilisée dans le même ordinateur, elle retourne la valeur zéro.

Mot clé associé: Aucun

**GOTO**

G <numéro de bloc>.<numéro de ligne>  
G <expression numérique>

>GOTO 11.95  
>GOTO 10.1+(BT/10)

COMMANDE: Se branche sur la ligne indiquée. Il est déconseillé d'utiliser cette commande qui rend trop fréquemment les programmes illisibles. Il est encore plus vivement déconseillé de sortir du bloc courant. En effet, si dans un bloc exécuté avec la commande DO se trouve une commande GOTO sortant de ce bloc, NODAL l'interprète comme une commande RETURN. Utilisez la commande DO.

Mot clé associé: Aucun

**GRAPH**

FONCTION: Cette fonction ainsi que toutes les instructions ayant trait au graphisme ne sont pas étudiées dans ce manuel. Voir PS-CO Note 79-14.

**GREEN**

GREEN

```
>WRITE(COLOUR)POSIT(26,11)GREEN"Hello World!"
```

VARIABLE CHAINE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur d'écriture (ou de fond si précédé de "BACKGR") des caractères suivants. Le fait de spécifier une couleur d'écriture définit automatiquement la couleur de fond comme noire. Une spécification de couleur de fond doit donc toujours suivre, et non précéder, la spécification de couleur d'écriture.

Mots clés associés: BACKGR, BLACK, BLUE, ERASE, INVERT, LBLUE, MAGENT, RED, WHITE, YELLOW

**HELP**

HELP

```
>HELP
<voir texte>
```

FONCTION: Le fait de taper HELP donne une liste des commandes NODAL avec leurs abréviations minimales.

Mot clé associé: Aucun

**IDEV**

```
IDEV=OPEN(<lecture>,<nom de fichier>)
```

```
>SET IDEV=OPEN("R","TERM-MODE")
>CLOSE(IDEV);SE IDEV=TERM-MODE
```

FONCTION: NODAL offre la possibilité de changer le canal d'entrée, normalement affecté au terminal. L'exemple ci-dessus ouvre le fichier TERM-MODE en lecture et interprète les commandes contenues dans ce fichier à chaque interrogation clavier.

Mots clés associés: ASK, \$ASK, CLOSE, ODEV, OPEN

**IF**

IF <expression logique>;<restant de la ligne>

>IF A<132;TYPE "A est au maximum égal à 131"

>IF A<=1;IF B<>2;SET C=3

>IF A>=1 OR B<>2;IF C=33;SET D=A\*B

COMMANDE: Détermine si l'expression logique est vraie pour exécuter, le cas échéant, le <restant de la ligne>. L'opérateur AND est obtenu en "cascadant" le nombre de IF nécessaires. Ce type de IF ne doit jamais être suivi par une parenthèse. Voir la commande suivante.

Mots clés associés: IF(, \$IF

**IF(**

IF(<expression numérique>)<voir texte>

>10.10 IF(A)3.2; DO 4

>10.10 IF(A)3.2,3.3,3.4

COMMANDE: Sélectionne une ligne à laquelle le programme doit se brancher en fonction du signe de l'<expression numérique>. Le premier branchement est effectué si l'<expression numérique> est négative, le second si elle est nulle et le troisième si elle est positive. L'<expression numérique> est estimée nulle si sa valeur absolue est inférieure 1E-6.

Mots clés associés: IF, \$IF

**IMAGTR**

IMAGTR(<tableau source>,<tableau objet>)

>IMAGTR(IMA1,IMA2)

FONCTION: Transforme un tableau contenant une image d'écran couleur en un tableau contenant une image d'écran noir et blanc ou vice-versa. Les tableaux contenant les images doivent être dimensionnés à 2087 entiers pour un écran couleur et à 1026 entiers pour un écran noir et blanc. La lecture d'un de ces tableaux est obtenue avec PIXIN, l'écriture avec PIXOUT.

Mots clés associés: PIXIN, PIXOUT

**IMEX**

IM(<ordinateur>)<liste de: commande>

>IMEX(MCR)FOR C=100,100,800:TYPE C BFC(C)!

COMMANDE: Cette commande permet la programmation interactive, à condition que la <liste de: commande> tienne sur la même ligne, et qu'aucun paramètre ne doive être retourné. L'exemple ci-dessus va faire exécuter le restant de la ligne dans l'ordinateur MCR. Si l'ordinateur appelé n'a pas répondu au bout de 0,6 seconde, la commande est abandonnée.

Mot clé associé: EXECUTE

**INT**

INT(<expression numérique>)

>TYPE INT(10/3)

3

FONCTION: Si l'<expression numérique> est positive, elle arrondit à l'entier inférieur, si négative, à l'entier supérieur. Ce qui fait que INT(-.9) donne comme résultat 0. Voir également la définition d'un entier au début du manuel. Attention: INT(2↑3) peut donner 7.

Mots clés associés: ABS, FPT

**INVERT**

WRITE(<unité logique>)INVERT<chaîne de caractères>

>WRITE(BANDW(2))POSIT(26,11)INVERT"Hello World!"

VARIABLE CHAINE: Permet, à l'intérieur d'une chaîne de caractères, d'inverser les couleurs d'écriture et de fond des caractères suivants. Reste actif jusqu'au prochain INVERT ou ERASE ou jusqu'à la prochaine indication de couleur.

Mots clés associés: BACKGR, BLACK, BLUE, ERASE, GREEN, LBLUE, MAGENT, RED, WHITE, YELLOW

**IOR**

IOR(<argument entier>,<argument entier>)

>SET C=IOR(A,B)

FONCTION: Effectue bit par bit l'opération booléenne OU entre les deux <arguments entiers>. Le résultat est bien sûr un entier.

Mots clés associés: AND, NEG

**KCURC**

KCURC(<colonne curseur>)

>SET H=KCURC

>SET KCURC=H

FONCTION: Cette fonction permet la lecture ou l'écriture de la position horizontale du curseur de l'écran sur lequel il est connecté. Si aucun curseur n'est connecté, l'erreur "DEVICE NOT CONNECTED" est affichée. La valeur de la position horizontale du curseur va de 1 à 64.

Mots clés associés: BALLST, KCURL

**KCURL**

KCURL(<ligne curseur>)

>SET V=KCURL

>SET KCURL=V

FONCTION: Cette fonction permet la lecture ou l'écriture de la position verticale du curseur de l'écran sur lequel il est connecté. Si aucun curseur n'est connecté, l'erreur "DEVICE NOT CONNECTED" est affichée. La valeur de la position verticale du curseur va de 1 à 32, mais seules les valeurs de 1 à 24 sont significatives, étant donné qu'il n'y a que 24 lignes affichées.

Mots clés associés: BALLST, KCURC

**KDSABL**

KDSABL(<unité logique>)

>KDSABL(COLOUR)

FONCTION: Déconnecte le curseur sur l'écran indiqué par l'<unité logique> ainsi que la boule. Tenter de déconnecter un curseur qui n'existe pas affiche le message "DEVICE NOT RESERVED".

Mots clés associés: BALLST, KENABL

**KENABL**

KENABL(<unité logique>)

>KENABL(BANDW(3))

FONCTION: Connecte le curseur sur l'écran indiqué par l'<unité logique>(le curseur apparait et peut être contrôlé à l'aide de la boule). Etant donné que seul un curseur à la fois peut être connecté, tenter de connecter un second curseur affiche le message "DEVICE ALREADY RESERVED".

Mots clés associés: BALLST, KDSABL

**KINIT**

KINIT(<voir texte>)

>SET KINIT(1,0,1E-38,1E-38,1,1,"mm","position")=1E-38

>SET KINIT(4,3, 1E38, 1E38,32567,32567,"mm","position")=1E38

FONCTION: Initialise les boutons (KNOBS). Les différents paramètres sont, dans l'ordre:

- Numéro du bouton (de 1 à 4)
- Mode (de 0 à 3): bit 0 : 0=mode cumulatif  
1=mode incrémental

bit 1 : 0=pas de touche validation

1=touche validation active

- Valeur réelle maximale autorisée
  - Valeur réelle minimale autorisée
  - Valeur minimale du poids d'un incrément (6 digits d'affichés)
  - Facteur d'échelle, vaut normalement 0.
  - Chaîne de 4 caractères max. indiquant l'unité affichée
  - Chaîne de 24 caractères max. indiquant le paramètre affiché
- La valeur à droite du signe "=" est la valeur initiale du paramètre. Celle-ci doit être comprise entre la valeur minimale et la valeur maximale indiquées par les 3ème et 4ème paramètres. Dans le cas où le mode incrémental à été choisi, la valeur initiale doit être mise à zéro.

Mots clés associés: KNBGR, KPURGE, KWRITE

**KNBGR**

KNBGR

>6.02 SE GR=0

>6.04 SE BT=MWAIT(GR);IF GR<>KNBGR;SE XI=1;RETURN

FONCTION: Cette fonction retourne le numéro du groupe de touches des "knobs", si l'utilisateur l'a préalablement réservé. Dans la cas contraire, le message "DEVICE NOT CONNECTED" est affiché.

Mots clés associés: BALLGR, TPGR, TRIGER

**KPURGE**

KPURGE(<numéro du bouton de 1 à 4>)

>KPURGE(3)

FONCTION: Annule la réservation de l'encodeur indiqué et efface la partie de l'écran associé.

Mot clé associé: KINIT

**KWRITE**

KWRITE(<numéro du bouton de 1 à 4>)=<chaîne de caractères>

>\$SE KWRITE(3)="Position"

FONCTION: Permet l'écriture d'une ligne de 23 caractères au maximum dans la zone concernée par le <numéro de bouton> indiqué.

Mot clé associé: KINIT

**LBLUE**

LBLUE

>WRITE(COLOUR)POSIT(26,11)LBLUE"Hello World!"

VARIABLE CHAÎNE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur d'écriture (ou de fond si précédé de "BACKGR") des caractères suivants. Le fait de spécifier une couleur d'écriture définit automatiquement la couleur de fond comme noire. Une spécification de couleur de fond doit donc toujours suivre, et non précéder, la spécification de couleur d'écriture.

Mots clés associés: BACKGR, BLUE, ERASE, GREEN, INVERT, MAGENT, RED, WHITE, YELLOW

**LDEF**

LD <liste de: nom de fichier>

>LDEF fich1 fich2 fich3

COMMANDE: Charge le(s) fichier(s) indiqué(s) dans la zone réservée aux fonctions définies par l'utilisateur. Les anciennes fonctions définies sont effacées.

Mots clés associés: DEFINE-, SDEF, ZDEF

**LEGEND**

LEGEND(<numéro de la touche de 1 à 16>)=<chaîne de caractères>  
 LEGEND(0)=<chaîne de 48 caractères max. pour le titre>

>\$SE LEGEND(16)="EXIT"  
 >\$SE LEGEND(0)="TITRE"

**FONCTION:**

- Avec le numéro de la touche égal à zéro:
  - Efface l'écran TPUSER et affiche la chaîne de caractères servant de titre, centrée et sur la première ligne.
- Avec le numéro de la touche compris entre 1 et 16:
  - Si la chaîne de caractères n'est pas vide:
    - Affiche la chaîne de caractères à l'intérieur de la touche concernée et valide cette touche. Le caractère "\" permet de sauter une ligne. La zone d'affichage d'une touche comprend 3 lignes de 8 caractères chacune. Le texte d'une ligne est automatiquement centré.
  - Si la chaîne de caractères est vide:
    - Efface et invalide la touche concernée.

Mot clé associé: LEGLUN

**LEGLUN**

LEGLUN

>SE LU.N=LEGLUN  
 >WRITE(LEGLUN)POSIT(18,7)"Hello World!"

**FONCTION:** Cette fonction permet l'écriture sur l'écran TPUSER. Ceci peut être utile si des messages doivent être détectés rapidement par l'opérateur. Une ligne d'écran TPUSER a une longueur de 48 caractères.

Mots clés associés: BANDW, COLOUR, GRAPH, LEGEND

**LGREAD**

LGREAD(<numéro de la touche>)

>\$SE LEGEND(BT)=INVERT LGREAD(BT) INVERT

**FONCTION:**

- Si le numéro de la touche est zéro:
  - Rend une chaîne de caractères contenant le titre.
- Si le numéro de la touche est différent de zéro:
  - Rend une chaîne de 24 caractères, correspondant au texte figurant sur la touche concernée. En effet, chaque touche comporte 3 lignes de 8 caractères.

L'exemple ci-dessus est couramment utilisé pour inverser une touche qui a été activée par l'utilisateur.

Mot clé associé: LEGEND

**LISA**

LISA

>LISA

<voir texte>

FONCTION: Affiche la liste des fonctions mathématiques et quelques symboles, par exemple: SQR, MOD, INT, FPT, ABS, SGN, AT2, SIN, COS, EXP, LOG, PIE, ARG, REAL, STRING, BLOCK.

Mot clé associé: Aucun

**LISC**

LISC

>LISC

<voir texte>

FONCTION: Affiche une liste des dénominations mnémoniques des ordinateurs accessibles par IMEX ou EXECUTE et de leurs numéros.

Mot clé associé: Aucun

**LISD**

LISD

>LISD

<voir texte>

FONCTION: Affiche une liste des fonctions définies par l'utilisateur, de leur taille et de leur(s) paramètre(s).

Mot clé associé: Aucun

**LISG**

LISG

>LISG

<voir texte>

FONCTION: Affiche la liste des variables globales.

Mots clés associés: CREATE, DELETE, FETCH, STORE

**LISR**

LISR

>LISR  
 <voir texte>

**FONCTION:** Affiche la liste des fonctions du système résident, avec leurs types et leurs spécifications de paramètres.

**Mot clé associé:** WHAT

**LIST**

Ll [<numéro de bloc>[.<numéro de ligne>]]  
 Ll <liste de: comme ci-dessus>

>LIST  
 <voir texte>  
 >LIST 1 2 3.4  
 <voir texte>

**COMMANDE:** Le premier exemple affiche la liste complète du programme actuellement dans la zone de travail. Le deuxième exemple n'affiche que les blocs 1, 2 et la ligne 40 du bloc 3.

**Mot clé associé:** Aucun

**LISV**

LISV

>LISV  
 <voir texte>

**FONCTION:** Affiche la liste des variables locales créées par l'utilisateur ainsi que l'espace encore disponible dans la zone de travail.

**Mots clés associés:** LISG, WHAT

**LOAD**

LO <nom du fichier>

>LOAD ABCDEF  
 >LOAD \$FN;% Indirection:FN est la var. contenant le nom du fichier

**COMMANDE:** Charge un programme NODAL dans la zone de travail. Les numéros de bloc/numéros de lignes du programme précédent se retrouvant dans le nouveau programme sont automatiquement écrasés. Les numéros de bloc/ numéros de lignes du programme précédent ne se retrouvant pas dans le nouveau programme sont ajoutés. Si ceci n'est pas voulu, voir la commande OLD.

**Mots clés associés:** OLD, RUN, SAVE

**LOCAL**

LOCAL

&gt;SE CN=LOCAL

FONCTION: Donne le numéro d'identification de l'ordinateur sur lequel cette fonction est appelée.

Mot clé associé: Voir LSTAT

**LOG**

LOG(&lt;expression numérique&gt;)

```
>TYPE LOG(9999)
      9.2102
```

```
>TYPE LOG(9999)/LOG(10)
      3.9999
```

FONCTION: Calcule le LOGarithme naturel d'une <expression numérique> (supérieure à 0). Le deuxième exemple montre l'obtention du LOGarithme décimal d'une expression numérique.

Mot clé associé: EXP

**LSTAT**

LSTAT(&lt;numéro d'identification d'un ordinateur&gt;)

```
>!F LSTAT(SYMBOL(CPS))=0;TYPE"Transmission vers CPS impossible"
```

FONCTION: Retourne 1 si le "link" vers l'ordinateur indiqué est ouvert et zéro dans le cas contraire.

Mot clé associé: Aucun

**MAGENT**

MAGENT

```
>WRITE(COLOUR)MAGENT"Hello World!"
```

VARIABLE CHAINE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur d'écriture (ou de fond si précédé de "BACKGR") des caractères suivants. Le fait de spécifier une couleur d'écriture définit automatiquement la couleur de fond comme noire. Une spécification de couleur de fond doit donc toujours suivre, et non précéder, la spécification de couleur d'écriture.

Mots clés associés: BACKGR, BLACK, BLUE, ERASE, GREEN, INVERT, LBLUE, RED, WHITE, YELLOW

**MAX**

MAX(<nom du tableau>)

>SE A=MAX(ABC)

FONCTION: Cette fonction trouve la valeur maximale contenue dans un tableau d'entiers ou de réels.

Mot clé associé: MIN

**MIN**

MIN(<nom du tableau>)

>SE B=MIN(ABC)

FONCTION: Cette fonction trouve la valeur minimale contenue dans un tableau d'entiers ou de réels.

Mot clé associé: MAX

**MOD**

MOD(<expression numérique 1>,<expression numérique 2>)

>TYPE MOD(10,3)

1

FONCTION: Calcule le MODULO, c'est-à-dire le reste de la division entière de l'<expression numérique 1> par l'<expression numérique 2>.

Mot clé associé: Aucun

**MWAIT**

MWAIT(<numéro du groupe>)

>SET GR=0

>SET BT=MWAIT(GR)

FONCTION: Donne à l'utilisateur la possibilité d'attendre la pression d'une touche. Dans l'exemple ci-dessus et lors de la pression sur une touche, retourne le numéro du groupe dans GR et reçoit le numéro de la touche dans BT.

Mot clé associé: EWAIT

**NEG**

NEG(<expression numérique entière>)

```
>TYPE NEG(0)
-1
>TYPE NEG(8)
-9
```

FONCTION: Cette fonction complète les 16 bits de l'<expression numérique entière>.

Mots clés associés: AND, IOR

**NODLIN**

NODLIN(<numéro de bloc>.<numéro de ligne>)

```
>$SET ST.RIN=NODLIN(10.1)
>$SET NODLIN(10.1)="TYPE\34HELLO WORLD!\34"
```

FONCTION: Cette fonction permet l'écriture ou la lecture des lignes NODAL elles-mêmes. Le premier exemple est la construction d'une chaîne de caractères ayant le contenu d'une ligne (sans les numéros de bloc et de ligne). Le second est la construction d'une ligne NODAL. IMPORTANT: Le <numéro de bloc>.<numéro de ligne> de la ligne construite doit être supérieur à la ligne qui en fait la construction.

Mot clé associé: Aucun

**ODEV**

ODEV=<open>(<"W"ou"R"ou"A">,<"nom du fichier de sortie">,<liste de: commande>)

```
>SE OD=ODEV
>SE ODEV=OPEN("W","LINE-PRINTER");LIST
>CLOSE(ODEV);SE ODEV=OD
```

FONCTION: NODAL offre la possibilité de changer le canal de sortie, normalement affecté au terminal. L'exemple ci-dessus dirige la liste du programme vers l'imprimante.

Mots clés associés: CLOSE, IDEV, OPEN

**OLD**

OL <nom du fichier>

>OLD ABCDEF

COMMANDE: Cette commande nettoie la zone de travail, charge le programme NODAL indiqué, puis rend la main au programmeur à travers l'éditeur NODAL. Cette commande n'est utilisée qu'en mode immédiat, c'est-à-dire sans être précédée d'un numéro de bloc/numéro de ligne.

Mots clés associés: LOAD, RUN, SAVE

**OPEN**

OP(<"R"ou"W"ou"A">,<nom du fichier>)

>SE F.NUM=OPEN("A","ABCDEF")

COMMANDE: Ouvre le fichier défini par le <nom de fichier> pour la lecture (Read), l'écriture (Write) ou pour ajouter (Append) des éléments à ce fichier et, dans l'exemple ci-dessus, retourne le numéro d'unité logique (LUN) dans F.NUM.

Mots clés associés: CLOSE, IDEV, ODEV, TYPE

**OVERLAY**

OV <nom de fichier>[<liste de: expression numérique (8 maximum)>]

>OVERLAY SPGM1 2,3,4

COMMANDE: Permet à un sous-programme d'être chargé et exécuté au cours d'un programme principal, sans gêner ce dernier. Le sous-programme peut utiliser les mêmes "n°de bloc/n°de ligne" et les mêmes noms de variables sans interférer avec le programme principal. Le programme principal et le sous-programme partagent temporairement la même zone de travail. La somme de leurs dimensions ne doit donc pas dépasser la dimension maximale de la zone de travail. Le sous-programme disparaît après utilisation; ainsi, le programme principal peut appeler d'autres sous-programmes. Un sous-programme peut en appeler un autre etc... Le passage de paramètres peut se faire en faisant suivre le <nom de fichier> par un maximum de 8 expressions numériques. Ces valeurs sont alors mises dans le tableau ARG. Dans l'exemple ci-dessus, ARG(1)=2, ARG(2)=3, ARG(3)=4. Un inconvénient de ce système est que les fautes rapportées le sont toujours au niveau du programme principal, à la ligne qui appelle le sous-programme, le message d'erreur, lui, décrivant de la manière habituelle la faute dans le sous-programme. La commande OVERLAY peut, par exemple, être utilisée pour gérer les parties fixes des affichages.

Mot clé associé: Aucun

**PIE**

PIE

```
>TYPE PIE
    3.1415
```

FONCTION: Fournit la valeur du rapport circonférence / diamètre du cercle. La valeur interne a 9 chiffres significatifs, mais la commande TYPE prend son format par défaut, c'est-à-dire avec quatre chiffres derrière la virgule.

Mot clé associé: Aucun

**PIXIN**

```
PIXIN(<numéro d'unité logique>,<nom du tableau d'entiers>)
```

```
>DIM-I IM.AGE(1026)
>PIXIN(BANDW(3),IM.AGE)
```

FONCTION: Cette fonction permet la sauvegarde d'une image d'écran (noir et blanc ou couleur) dans un tableau d'entiers. Ce tableau doit être dimensionné à 1026 pour un écran noir et blanc et à 2087 pour un écran couleur.

Mots clés associés: IMAGTR, PIXOUT

**PIXOUT**

```
PIXOUT(<numéro d'unité logique>,<nom du tableau d'entiers>)
```

```
>PIXOUT(BANDW(1),IM.AGE)
```

FONCTION: Cette fonction restitue une image préalablement sauvée avec PIXIN. La conversion d'écran couleur à écran noir et blanc ou vice-versa peut être obtenue avec la fonction IMAGTR.

Mots clés associés: IMAGTR, PIXIN

**POSIT**

```
POSIT(<position horizontale>,<position verticale>)
```

```
>WRITE(COLOUR)POSIT(1,1)"Origine"
>WRITE(COLOUR)POSIT(62,25)"fin"
```

FONCTION: Cette fonction permet de définir l'endroit où sera écrit le premier caractère de la chaîne de caractères à afficher. Dans le premier exemple, la lettre "O" sera affichée dans le coin en haut et à gauche, alors que dans le second, la lettre "N" sera affichée dans le coin en bas et à droite.

Mot clé associé: Aucun

**QUIT**

Q

&gt;QUIT

COMMANDE: Cette commande sort de NODAL pour passer sous le contrôle du Système d'Exploitation (SINTRAN-III).

Mot clé associé: Aucun

**RED**

RED

&gt;WRITE(COLOUR)RED"Hello world!"

VARIABLE CHAINE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur d'écriture (ou de fond si précédé de "BACKGR") des caractères suivants. Le fait de spécifier une couleur d'écriture définit automatiquement la couleur de fond comme noire. Une spécification de couleur de fond doit donc toujours suivre, et non précéder, la spécification de couleur d'écriture.

Mots clés associés: BACKGR, BLACK, BLUE, ERASE, GREEN, INVERT, LBLUE, MAGENT, WHITE, YELLOW

**REMIT**

REM <liste de: nom de tableau et/ou de variable>

&gt;REMIT A B CO

COMMANDE: Lors d'une commande EXECUTE, il est souvent nécessaire de retourner des valeurs obtenues dans l'autre ordinateur. La commande REMIT, qui termine alors la sous-tâche effectuée dans l'autre ordinateur, permet de retourner ces valeurs sous forme de tableaux et/ou de variables. Même si la sous-tâche ne contient pas de commande REMIT, NODAL effectue cette commande à la fin de la sous-tâche ou si une faute est rencontrée mais, dans ce cas, ne rend ni les valeurs, ni le code d'achèvement (completion code). Dans l'exemple ci-dessus, et si aucune faute n'est rencontrée, le code d'achèvement CO permet de vérifier si l'opération s'est déroulée correctement.

Mot clé associé: EXECUTE

**RETURN**

RET

```
>1.10 DO 3;% Initialisation
>1.99 END
```

```
-----
>3.10 SE CO=0;$SE MY.NAM="Fred"
>3.20 DO 3.80
>3.30 RETURN
>3.80 FOR I=1,SIZE(MY.NAM):TY SUBS(I,I,MY.NAM)!
>RUN
F
r
e
d
```

COMMANDE: Normalement, une commande DO <numéro de bloc> se termine à la fin de la dernière ligne de ce bloc. La commande RETURN permet de rompre cette habitude en retournant plus tôt à l'instruction qui suit le DO appelant. Ceci peut servir à libérer de la place dans les numéros de ligne les plus élevés, afin de pouvoir y loger de petites routines qui tiennent sur une seule ligne, et que l'on peut alors appeler avec DO <numéro de bloc>.<numéro de ligne>.

Mot clé associé: DO

**ROF**

RO

```
>FOR I=32,127;$SE A=\I;$IF A="B";ROF
>TYPE I
66
```

COMMANDE: Une boucle FOR se termine normalement lorsque la valeur de la dernière <expression numérique> est atteinte. Si ceci n'est pas voulu, il est possible de forcer la fin de la boucle avec la commande ROF. Si plusieurs boucles FOR sont imbriquées sur la même ligne, une seule commande ROF permet de sortir de toutes les boucles.

Mot clé associé: FOR

**RUN**

RU[[<numéro de bloc>[.<numéro de ligne>]][<nom du fichier>]

```
>RU
>RU[2.7]
>RU ABCDEF
>RU[5]ABCDEF
```

COMMANDE: Dans le premier et le second exemple, le programme doit être dans la zone de travail, donc soit qu'il vienne d'être saisi, soit qu'il vienne d'être chargé (avec LOAD ou OLD). Il suffit alors de taper RUN pour le lancer. Le deuxième exemple démarre le programme à partir du n<sup>o</sup> de bloc/n<sup>o</sup> de ligne 2.7, l'exemple suivant charge le programme ABCDEF dans la zone de travail avant de le lancer. Quant au dernier exemple, il charge le programme puis le lance à partir du début du bloc 5. Cette commande nécessite l'emploi de parenthèses carrées autour des numéros de bloc/ligne.

Mots clés associés: LOAD, OLD, RUN, SAVE

**SAVE**

SA <nom du fichier>[<liste de:n<sup>o</sup> de bloc et/ou n<sup>o</sup> de bloc/n<sup>o</sup> de ligne>][<liste de: nom de tableau>]

SA <nom du fichier>[<tout le programme>][<toutes les variables>]

SA"<nom du fichier>"

```
>SAVE ABCDEF
>SAVE ABCDEF 2 3 TAB
>SAVE ABCDEF TAB
>SAVE ABCDEF ALLP ALLV
>SAVE"ABCDEF"
```

COMMANDE: Sauvegarde le programme et/ou les variables sur le disque dur de l'ordinateur sur lequel vous travaillez. Le premier exemple sauvegarde le programme, le deuxième ne sauvegarde que les blocs 2 et 3 ainsi que le tableau TAB, le troisième, que le tableau TAB, le suivant, tout le programme et toutes les variables. Le dernier exemple permet de créer un fichier sans passer par le système d'exploitation SINTRAN, et ne doit donc être utilisé que la première fois, suivi ou non par les mêmes options que celles que nous venons de voir ci-dessus. A noter que l'absence d'option correspond au suffixe ALLP.

Mots clés associés: LOAD, OLD, RUN

**SDEF**

SD"<nom de fichier>"  
SD <nom du fichier>

>SDEF"ABC:DF"  
>SDEF ABC:DF

COMMANDE: Sauve les fonctions définies par l'utilisateur dans le fichier spécifié. Le fichier doit d'abord avoir été créé, par exemple en entourant la première fois son nom de guillemets.

Mots clés associés: DEFINE-, LDEF, ZDEF

**SDMIN**

SDMIN(<nom du tableau d'entiers>)

>DIM-1 CARACT(1027)  
>SDMIN(CARACT)

FONCTION: Cette fonction permet la sauvegarde des caractères spéciaux créés par l'utilisateur (par exemple avec le programme GENECAR) dans un tableau d'entiers dimensionné au minimum à 1027. Ne concerne que les écrans noir et blanc et couleur.

Mot clé associé: SDMOUT

**SDMOUT**

SDMOUT(<nom du tableau d'entiers>)

>SDMOUT(CARACT)

FONCTION: Cette fonction permet la restitution de caractères spéciaux, ceux-ci ayant du être au préalable créés (par exemple avec le programme GENECAR) et sauvés (avec SDMIN) dans le tableau d'entiers. Ne concerne que les écrans noir et blanc et couleur.

Mot clé associé: SDMIN

**SET**

SE <nom de la variable ou de la fonction>=<expression numérique>

>SET RI.EN=0;SE UN=1;SE CO=0

COMMANDE: Affecte une valeur numérique à une variable ou à une fonction.

Mot clé associé: \$SET

**SGN**

SGN(<expression numérique>)

>TYPE SGN(4-9)  
-1

FONCTION: Donne le signe de l'<expression numérique>:  
+1 si nul ou positif  
-1 si négatif

Mot clé associé: ABS

**SHIFT**

SHIFT(<expression numérique entière>,<n>) voir texte

>SE A=4;SE A=SHIFT(A,1);TYPE A  
8

FONCTION: La valeur de l'expression numérique doit être comprise entre +32767 et -32768. Si la valeur <n> est positive, le décalage se fait à gauche. Si <n> est négatif, à droite. Les bits perdus sont remplacés par des zéros.

Mot clé associé: Aucun

**SIN**

SIN(<expression numérique>)

>TYPE SIN(PIE/4)  
.7071

FONCTION: Donne le SINUS de l'<expression numérique>, qui doit être exprimée en Radians.

Mots clés associés: COS, AT2, PIE

**SIZE**

SIZE(<chaîne de caractères>)

>\$SE A=" Anticonstitutionnellement"  
>TYPE SIZE(A)  
27

FONCTION: Donne le nombre de caractères de la chaîne, espaces compris.

Mot clé associé: Aucun

**SORT**

**SORT**(<nom du tableau de chaînes de caractères>,<+ ou ->)

>**SORT**(TEL,+)

**FONCTION**: Trie un tableau de chaînes de caractères par ordre ASCII croissant (+) ou décroissant (-).

Mot clé associé: Aucun

**SQR**

**SQR**(<expression numérique>)

>**TYPE SQR**(9)

3

**FONCTION**: Calcule la racine carrée de l'<expression numérique>.

Mot clé associé: Aucun

**SQUEEZ**

**SQUEEZ**

>**SQUEEZ**

**FONCTION**: Cette fonction permet de compacter le programme NODAL en remplaçant le nom des commandes par leur abréviation minimale, en éliminant les espaces inutiles et en supprimant les commentaires. Elle permet donc de gagner de la place dans la zone de travail en accélérant la vitesse d'exécution. La perte des commentaires est sans importance si un fichier symbolique (c'est-à-dire un fichier ASCII) a servi à écrire le programme.

Mot clé associé: Aucun

**START**

**START**("[\*]<nom de fichier SINTRAN>"[,CO])

>**START**("\*<CPS>(C-S)TVPR",CO)

**FONCTION**: Démarre un processus ("R" program" dans le contexte SINTRAN). Ce programme peut être écrit en n'importe quel langage. Le signe "\*" est spécifique aux Consoles: il indique l'exécution d'une classe spéciale de processus dénommé "switch-process". Cette classe permet l'accès aux ressources du "MIP". Pour de plus amples renseignements, voir la note de B.Carpenter: Software Process Manipulation Package for Sintran 3 (PS-CO-Note 80-31).

Mot clé associé: Aucun

**STORE**

STORE(<nom de la variable>,<nom de la variable globale>,CO)

>STORE(A,"ABC",CO)

FUNCTION: Cette fonction permet de transférer le contenu d'une variable dans une variable globale.

Mots clés associés: CREATE, DELETE, FETCH, LISC

**STRARG**

STRARG="<chaîne de caractères>"

>\$SET S=STRARG

FUNCTION: Il n'y a qu'une seule variable globale du type chaîne, qui ne supporte pas plus de 40 caractères. Cette variable peut, par exemple, servir à passer un nom de fichier d'une tâche à une autre.

Mots clés associés: ARG, OVERLAY, START

**SUBS**

SUBS(<expr. numérique 1>,<expr. numérique 2>,<chaîne de caractères>)

>\$SE ST.RIN="CALIFORNIA"

>TYPE SUBS(5,7,ST.RIN)

FOR

>\$SE SUBS(5,7,ST.RIN)="couleur";TYPE ST.RIN

CALICOUNIA

FUNCTION: Permet de modifier une chaîne de caractères. Dans le premier exemple ci-dessus, les caractères de 5 à 7 de la chaîne sont affichés. Dans le second, la chaîne est modifiée par remplacement des caractères de 5 à 7 par un même nombre de caractères de la chaîne incidente.

Mot clé associé: Aucun

**TIME**

TIME

>TYPE TIME

56096.2

FUNCTION: Donne le temps interne en secondes.

Mot clé associé: DATE

**TPGR**

TPGR

&gt;9.10 SE GR=0

&gt;9.20 SE BT=MWAIT(GR);IF GR&lt;&gt;TPGR;SE XI=1;RETURN

FUNCTION: Cette fonction est, en fait, une variable système qui retourne la valeur du groupe d'événements associé aux touches du TPUSER.

Mots clés associés: BALLGR, KNBGR, TRIGER

**TRANS**

TRANS(<ordinateur source>,<fichier source>,<ordinateur objet>,<fichier objet>,<code d'erreur>)

&gt;TRANS(CONS4,(OP-GE)ABC,CPS,(C-S)DEF,CO)

FUNCTION: Cette fonction permet la transmission d'un fichier d'un ordinateur à l'autre. Le <code d'erreur> indique l'erreur due au processus de transmission, et non l'erreur pouvant exister pendant la transmission elle-même. Cette fonction n'est pas implémentée sur l'ordinateur PRDEV.

Mot clé associé: Aucun

**TRIGER**

TRIGER

&gt;5.10 SE GR=0

&gt;5.20 IF GR=TRIGER;SE XI=1;RET

FUNCTION: Cette fonction est en fait une variable système qui définit la touche BACK ou HOME.

Mots clés associés: BALLGR, KNBGR, TPGR

**TVASK**

TVASK(<numéro d'unité logique>,<ligne>,<colonne>,<longueur>)

&gt;SE VA.LUE=TVASK(COLOUR,10,33,4)

FUNCTION: Permet la saisie, dans l'écran défini par le <numéro d'unité logique> et dans la fenêtre définie par les paramètres <ligne>, <colonne> et <longueur>, d'une valeur. Valeurs par défaut des paramètres:

    ligne : position du curseur  
    colonne : position du curseur  
    longueur : jusqu'à la fin de la ligne

Mot clé associé: TVEDIT

**TVEDIT**

TVEDIT(<numéro d'unité logique>,<ligne>,<colonne>,<longueur>)

```
>TVEDIT(BANDW(2),12,44,5)
>$SE ST.RIN=TVEDIT(COLOUR,10,33,4)
```

FONCTION: Permet la saisie, dans l'écran défini par le <numéro d'unité logique> et dans la fenêtre définie par les paramètres <ligne>, <colonne> et <longueur>, d'une chaîne de caractères. Valeurs par défaut des paramètres:

    ligne : position du curseur  
    colonne : position du curseur  
    longueur : jusqu'à la fin de la ligne

Mot clé associé: TVASK

**TVREAD**

TVREAD(<numéro d'unité logique>,<ligne>)

```
>$SE S.LIN=TVREAD(COLOUR,12)
```

FONCTION: Cette fonction permet de lire une chaîne de caractères affichée sur un écran noir et blanc ou couleur.

Mot clé associé: WRITE

**TWAIT**

TWAIT(<delai en secondes>)

```
>TWAIT(3.3)
```

FONCTION: Cette fonction permet d'attendre un temps donné par le <delai en secondes>. En fait, c'est un temps minimum, car il dépend du contexte au moment souhaité de la reprise en main. Délai minimum: 0.02 seconde. Imprécision: 0.02 seconde.

Mot clé associé: WAIT-TIME

**TYPE**

T[<liste de: article à imprimer>]

```
>1.10 $SE A="Albert";SE B=900.99;SE C=50
>1.20 TYPE A" a perdu"%4 B-C" Francs au jeu."!"Tant pis!"
>RUN
Albert a perdu 850 Francs au jeu.
Tant pis!
```

COMMANDE: Cette commande permet de sortir des informations dans un fichier ou sur l'unité de sortie (ODEV) de travail (par défaut l'écran écho). Elle permet la sortie de chaînes de caractères, de valeurs numériques, de caractères spéciaux de contrôle. Afin de pouvoir formater votre affichage, un paragraphe "Caractères Spéciaux" est à votre disposition au début de ce manuel. A noter,

dans l'exemple, l'arrondi produit par le format %4.

Mots clés associés: ODEV, WRITE

## VALUE

```
V <liste de: variables numériques>
>1.1 VALUE A+B
>DEF-FUN SOM(V-A,V-B)
-----
>TYPE SOM(2,3)
      5
```

COMMANDE: Cette commande termine l'écriture d'une fonction numérique. Dans l'exemple très simple ci-dessus, l'utilisateur a créé une fonction SOM, qui ne fait que... la somme des deux valeurs passées à cette fonction.

Mots clés associés: DEFINE-, \$VALUE

## WAIT

WA(<ordinateur>)

```
>2.10 EXECUTE(CPS) 3 CO;WAIT(CPS)
>2.15 IF CO<>0;SE XI=1;RETURN;% XI est le "flag" pour "exit"
```

COMMANDE: Lors de l'exécution d'une tâche dans un autre ordinateur, il faut attendre que ce dernier ait retourné les variables et/ou tableaux demandés. Ceci est fait par la commande WAIT.

Mots clés associés: EXECUTE, IMEX

## WAIT-TIME

WAIT-TIME <délai en secondes>

```
>WAIT-TIME 3.2
```

FONCTION: Cette fonction permet d'attendre un temps donné par le <délai en secondes>. En fait, il s'agit d'un temps minimum déterminé par le contexte au moment souhaité de la reprise en main. Délai minimum: 0.02 seconde. Imprécision: 0.02 seconde.

Mot clé associé: TWAIT

**WHAT**

WHAT(<nom de la variable ou du tableau>)

>WHAT(ABC)  
<voir texte>

FONCTION: Affiche les informations concernant une fonction, une variable ou un tableau cité, en utilisant le même format que LISR.

Mot clé associé: LISR

**WHILE**

WH<expression logique>;<commandes et/ou fonctions>

>SE A=0  
>WHILE A<999;SE A=A+1

COMMANDE: Cette commande, qui signifie "aussi longtemps que" permet de répéter une suite de commandes et/ou de fonctions jusqu' ce que l'<expression logique> soit vraie.

Mots clés associés: FOR

**WHITE**

WHITE

>WRITE(COLOUR)POSIT(26,11)WHITE"Hello World!"

VARIABLE CHAÎNE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur d'écriture (ou de fond si précédé de "BACKGR") des caractères suivants. Le fait de spécifier une couleur d'écriture définit automatiquement la couleur de fond comme noire. Une spécification de couleur de fond doit donc toujours suivre, et non précéder, la spécification de couleur d'écriture.

Mots clés associés: BACKGR, BLACK, BLUE, ERASE, GREEN, INVERT, LBLUE, MAGENT, RED, YELLOW

**WRITE**

WRITE(<écran>)[POSIT(<colonne>,<ligne>)][<couleurs>]<liste de: particle à imprimer>

>WRITE(COLOUR)POSIT(24,12)RED"Hello World!"  
>WRITE(BANDW(1))POSIT(26,12)\12"COUCOU"

FONCTION: Cette fonction permet d'afficher du texte sur un écran autre que le terminal ou l'écran graphique. D'autre part, WRITE obéit aux Codes de Contrôle décrits à la fin de ce manuel, mis à part les quatre écrans BANDW (c'est-à-dire black and white) qui afficheront en dégradé de gris à la place des différentes couleurs. Outre le Code ASCII standard, des caractères graphiques et spéciaux sont également disponibles.

Mots clés associés: TYPE, TVREAD

## YELLOW

YELLOW

>WRITE(COLOUR)POSIT(26,11)YELLOW"Hello World!"

VARIABLE CHAINE: Permet, à l'intérieur d'une chaîne de caractères, de définir la couleur d'écriture (ou de fond si précédé de "BACKGR") des caractères suivants. Le fait de spécifier une couleur d'écriture définit automatiquement la couleur de fond comme noire. Une spécification de couleur de fond doit donc toujours suivre, et non précéder, la spécification de couleur d'écriture.

Mots clés associés: BACKGR, BLACK, BLUE, ERASE, GREEN, INVERT, LBLUE, MAGENT, RED, WHITE

## ZDEF

Z

>ZDEF

COMMANDE: Vide la zone des fonctions définies par l'utilisateur.

Mots clés associés: DEFINE-, LDEF, SDEF

## \$ASK

\$A[<chaîne de caractères>]<liste de: nom de variable> voir texte

>\$ASK"Oui ou Non"RE.PON

Oui ou Non:\_

COMMANDE: Cette commande affiche d'abord la <chaîne de caractères> suivie d'un double point. L'utilisateur n'a plus qu'à saisir le contenu d'une/des variable(s) du type <chaîne de caractères> au clavier. Cette opération peut être répétée autant de fois que nécessaire derrière une seule commande \$ASK, mais toujours dans la limite des 80 caractères par ligne maximum.

Mot clé associé: ASK

**\$DO**

\$D <liste de: chaîne de caractères et/ou variable>

>SE A=1;SE B=21

>\$SE CL.COL="FOR I="A","B";WRITE(COLOUR)POSIT(1,I)&64"

>\$DO CL.COL;% Efface la zone utile de l'écran couleur

>FOR I=1,3;\$DO"LOAD DATA-"%1 I

COMMANDE: Cette puissante commande permet l'exécution d'une <chaîne de caractères> comme s'il s'agissait d'une ligne NODAL, cette chaîne pouvant être entrecoupée de variables de tous types. Le deuxième exemple peut être utilisé pour charger des fichiers.

Mot clé associé: DO

**\$IF**

\$I<expression logique à base de chaînes de caractères>;<restant de la ligne>

>\$IF RE.PON="OUI"OR RE.PON="NON";DO 64

COMMANDE: Cette commande est utilisée pour comparer deux chaînes de caractères. En cas de conformité, le restant de la ligne est effectué.

Mot clé associé: IF, \$MATCH

**\$MATCH**

\$M<chaîne de caractères 1><chaîne de caractères 2>[<si non>;  
<restant de la ligne>

>10.20 \$MATCH RE.PON"OUI":10.90;DO 12

>10.30 \$MATCH <REP.1 REP.2>"NONOUI":10.90;DO 12

COMMANDE: Cette commande permet de comparer deux chaînes de caractères. Si la <chaîne de caractères 1> contient la <chaîne de caractères 2>, le <restant de la ligne> est effectué. Dans le cas contraire c'est la partie <si non> située derrière le double point qui sera effectuée. Dans le premier exemple, si RE.PON contient OUI, la commande DO12 est effectuée. Dans le cas contraire, on saute à la ligne 10.90. Le deuxième exemple est identique au premier, sauf en ce qui concerne la première chaîne de caractères, qui a ici été remplacée par une concaténation.

Mot clé associé: \$IF

**\$PATTERN****\$P**

COMMANDE: Cette commande et ses options ne sont pas étudiées dans ce manuel, l'expérience montrant le faible taux d'utilisation de celles-ci. Pour de plus amples informations, voir "The Nodal System for the SPS".

**\$SET**

**\$S** <nom de la variable>=<liste de: chaîne de caractères/variable alphanumérique>

```
>$SET A="Anticonstitutionnellement"
>3.10 $SET B=INPUT(65):3.9;DO 13
>$SE PI.E=PIE
```

COMMANDE: Affecte une chaîne de caractères à une variable ou à une fonction. Le deuxième exemple est utilisé en cas d'entrées de données en provenance d'un fichier séquentiel: dans le cas de la rencontre de la fin de fichier (EOF), un saut en 3.9 est effectué. Dans le cas contraire, le restant de la ligne (ici la commande DO13) est effectuée. Le troisième exemple montre la conversion d'une valeur numérique en chaîne de caractères.

Mots clés associés: EVAL, SET

**\$VALUE**

**\$V** <liste de: variable chaîne de caractères>

```
>1.1 $VALUE"Ceci est : "A
>DEF-STR RESULT(S-A)
-----
>$SE A="vrai";TYPE RESULT(A)
Ceci est : vrai
```

COMMANDE: Cette commande termine l'écriture d'une fonction du type "DEFINE-STR". Dans l'exemple ci-dessus, l'utilisateur a créé une fonction RESULT, qui ne fait qu'une concaténation de deux chaînes de caractères.

Mots clés associés: DEFINE-, VALUE

%

%&lt;commentaire&gt;

&gt;1.10 % Titre

&gt;1.20 SET NJ.A=365;% Nombre de Jours par An

COMMANDE: Permet l'adjonction de commentaires. A ne pas confondre avec le symbol % dans une instruction de sortie. Il s'agirait, dans ce cas, d'un caractère spécial de formatage. Voir "Caractères Spéciaux", au début de ce manuel.

Mot clé associé: Aucun

**?ON - ?OFF**

?ON

?OFF

&gt;12.11 ?ON

&gt;12.20 SE I=I+1

&gt;12.56 ?OFF

COMMANDE: Permet de suivre l'exécution d'un programme par l'affichage de chaque ligne exécutée. Les numéros de bloc et de ligne ne sont pas indiqués.

Mot clé associé: Aucun

\*\*\*\*\*  
 \* CODE A.S.C.I.I. \*  
 \*\*\*\*\*

a) Codes de contrôle des "Drivers" des écrans monochromes et couleurs.

0 =	[CTRL]@	NUL	Ecriture en ROUGE et en majuscule sur fond NOIR
1 =	"	A SOH	" VERTE "
2 =	"	B STX	" BLEU FONCE "
3 =	"	C ETX	" BLANC "
4 =	"	D EOT	" BLEU CLAIR "
5 =	"	E ENQ	" VIOLET "
6 =	"	F ACK	" JAUNE "
7 =	"	G BEL	" NOIR "
8 =	"	H BS	Curseur en haut et à gauche
9 =	"	I TAB	(1-64) Position horizontale du curseur
10 =	"	J LF	Curseur une ligne vers le bas
11 =	"	K VF	(1-32) Position verticale du curseur
12 =	"	L FF	Efface l'écran et exécute les codes 3, 8, 20 et 30
13 =	"	M CR	Curseur en début de ligne (Carriage Return)
14 =	"	N SO	Echange couleur écriture/fond
15 =	"	O SI	Echange couleur écriture/fond (compatibilité SPS)
16 =	"	P DLE	(0-7) couleur du fond
17 =	"	Q DC1	(n) permet l'affichage des symboles dynamiques
18 =	"	R DC2	Clignotement arrêt
19 =	"	S DC3	Clignotement marche
20 =	"	T DC4	Sauver posit.curseur, couleurs d'écrit.et de fond
21 =	"	U NAK	Rendre posit.curseur, couleurs d'écrit.et de fond
22 =	"	V SYN	- Ignoré
23 =	"	W ETB	- "
24 =	"	X CAN	- "
25 =	"	Y EM	Curseur à droite
26 =	"	Z SUB	Curseur à gauche
27 =	"	[ ESC	- Ignoré
28 =	"	\ FS	Curseur vers le haut
29 =	"	] GS	Curseur vers le bas (Identique à 10)
30 =	"	↑ RS	Basculement en majuscule
31 =	"	_ US	Basculement en minuscule

## b) Code des caractères

32 = espace	64 = @	96 = `
33 = !	65 = A	97 = a
34 = "	66 = B	98 = b
35 = #	67 = C	99 = c
36 = \$	68 = D	100 = d
37 = %	69 = E	101 = e
38 = &	70 = F	102 = f
39 = '	71 = G	103 = g
40 = (	72 = H	104 = h
41 = )	73 = I	105 = i
42 = *	74 = J	106 = j
43 = +	75 = K	107 = k
44 = ,	76 = L	108 = l
45 = -	77 = M	109 = m
46 = .	78 = N	110 = n
47 = /	79 = O	111 = o
48 = 0	80 = P	112 = p
49 = 1	81 = Q	113 = q
50 = 2	82 = R	114 = r
51 = 3	83 = S	115 = s
52 = 4	84 = T	116 = t
53 = 5	85 = U	117 = u
54 = 6	86 = V	118 = v
55 = 7	87 = W	119 = w
56 = 8	88 = X	120 = x
57 = 9	89 = Y	121 = y
58 = :	90 = Z	122 = z
59 = ;	91 = [	123 = {
60 = <	92 = \	124 =
61 = =	93 = ]	125 = }
62 = >	94 = ^	126 = ~
63 = ?	95 = _	127 = delete

Caractères de Contrôle utilisables en Mode EDITion.  
 \*\*\*\*\*

Supposons que vous êtes dans l'éditeur de NODAL c'est-à-dire que vous avez devant vous l'écran ECHO, qui affiche:

>\_

...vous pouvez commencer à saisir votre programme (Le trait derrière la parenthèse angulaire représente le curseur):

>1.10 % MONPROGRAMME 22/01/88

>\_

...et là, vous vous rendez compte que vous vouliez écrire "MONPROGRAMME". Il y a plusieurs solutions à ce problème:

- ressaisir toute la ligne 1.10
- ou saisir, en tapant à la ligne suivante:

>EDIT 1.10,L

...ce qui aura pour effet de vous afficher la ligne 1.10 et de vous mettre en Mode EDITion:

1.10 % MONPROGRAMME 22/01/88

-

...tapez alors [CTRL]C, ce qui vous permettra de recopier un caractère de l'ancienne ligne dans la nouvelle. Tapez [CTRL]C jusqu'au caractère "G" inclus. Tapez alors dans l'ordre [CTRL]E R [CTRL]E [CTRL]D. L'effet de ces quatre caractères est le suivant:

- Début Mode Insertion
- La lettre R qui manque
- Fin Mode Insertion
- Copie jusqu'à la fin de l'ancienne ligne et sortie du Mode EDITion

Ceci n'est pas la solution la plus rapide pour modifier la ligne erronée. Cherchez donc, grâce au tableau ci-dessous, la meilleure solution.

[CTRL]A Efface le caractère précédent

- \* C Copie un car. de l'ancienne ligne dans la nouvelle ligne
- \* S Efface un caractère de l'ancienne ligne.(\*)
- \* E Débute (<) ou termine (>) le Mode Insertion
- \* H Copie l'ancienne ligne jusqu'à la fin
- \* Y Rajoute le restant de l'ancienne ligne à la nouvelle ligne
- \* M Equivaut à [RETURN]
- \* D Copie l'ancienne ligne jusqu'à la fin et sort du Mode EDITion
- \* O C Copie l'ancienne ligne jusqu'au caractère C non inclus
- \* Z C Copie l'ancienne ligne jusqu'au caractère C inclus
- \* P C Efface l'ancienne ligne jusqu'au caractère C non inclus
- \* X C Efface l'ancienne ligne jusqu'au caractère C inclus
- \* W Efface le dernier mot, affiche \
- \* Q Efface jusqu'au début de l'ancienne et de la nouvelle ligne.(\*)
- \* V C Prend le caractère C littéralement, même s'il s'agit d'un Code de Contrôle
- \* R Refrappe rapide
- \* T Refrappe alignée

- L Termine le Mode EDITion, identique à [RETURN]
- F Copie le restant de l'ancienne ligne, comme [CTRL]D, mais sans affichage

(\* Ces commandes ont une signification différente sur le PRDEV.

=====  
 = Bibliographie =  
 =====

- The Nodal System for the SPS : M.C.Crowley-Milling - G.C.Shering  
CERN SPS 78-07.
- Cookbook 4.4 : R.Cailliau - B.Carpenter PS/CO 08-84.
- Introduction to Application Program Writing - Cl.-H.Sicard  
PS/CO 86-001.
- User's Guide for Console Computer Nodal User Fonctions Running  
under SINTRAN III System - A.Gagnaire - F.Perriolat - C.Vicent  
PS/CO 79-11.
- Spécifications du package graphique des consoles principales  
PS/CO 79-14 DU 29/06/79.

=====  
 = REMERCIEMENTS =  
 =====

Mes remerciements à :

- G.Cuisinier
- A.Gagnaire
- D.Gueugnon
- B.Vandorpe

pour leur collaboration, et un remerciement anticipé à tous ceux  
 qui, par leurs remarques constructives, contribueront à la pro-  
 chaine version de ce manuel.

R.Hoh

Liste Des Commandes & Fonctions incluses dans ce Manuel

=====

abs	do	imagtr	lstat	shift
aeqm	edit	imex	magent	sin
and	emmes	int	max	size
arg	end	invert	min	sort
arsize	endpg	ior	mod	sqr
ascii	eqm	kcurc	mwait	squeez
ask	era	kcurl	neg	start
at2	erase	kdsabl	nodlin	store
backgr	ermes	kenabl	odev	strarg
ballgr	error	kinit	old	subs
ballst	eval	knbgr	open	time
bandw	ewait	kpurge	overlay	tpgr
bit	execute	kwrite	pie	trans
black	exist	lblue	pixin	trigger
blue	exp	ldef	pixout	tvask
butts	fcopy	legend	posit	tvedit
call	fetch	leglun	quit	tvread
close	first	lgread	red	twait
clzon	for	lisa	remit	type
colour	fpt	lisc	return	value
copy	fromc	lisd	rof	wait
cos	goto	lisg	run	wait-time
create	graph	lisr	save	what
date	green	list	sdef	while
def-call	help	lisv	sadmin	white
def-function	idev	load	sdmout	write
def-string	if	local	set	yellow
delete	if(	log	sgn	zdef
dimension				
\$ask	\$if	\$pattern	\$value	?off
\$do	\$match	\$set	%	?on

Caractères Graphiques des Ecrans Couleur et Noir & Blanc  
=====

128		129	-
130	┌	131	└
132	├	133	┤
134	▄	135	▀
136	┆	137	┇
138	┆	139	┆
140	┆	141	▀
142	▀	143	▀
144	▀	145	▀
146	▀	147	▀
148	▀	149	▀
150	▀	151	▀

Caractères Graphiques de "LEGLUN" (TP-USER)  
=====

128	-	129	-
130	▬	131	▬
132	▬	133	▬
134	▬	135	▬
136	▬	137	▬
138	┌	139	┌
140	└	141	└
142	┌	143	┌
144	▬	145	▬
146	▬	147	▬
148	┌	149	┌
150	└	151	└

Distribution

Groupe OP  
J.P. Potier  
F. Perriollat

Distribution (du résumé)  
Personnel Scientifique PS