UNIVERSITE DE NANCY FACULTE DES SCIENCES

CONTRIBUTION A LA CONSTRUCTION

D'UN COMPILATEUR ALGOL POUR I.B.M. 1620

THESE

pour l'obtention du

DOCTORAT de SPECIALITE MATHEMATIQUES (3ème cycle)

Soutenue devant le Jury le 22 février 65

par

Jacques ANDRE

Jury : Monsieur LEGRAS Président

Monsieur DEPAIX

Monsieur PAIR Examinateurs

UNIVERSITE DE NANCY FACULTE DES SCIENCES

CONTRIBUTION A LA CONSTRUCTION

D'UN COMPILATEUR ALGOL POUR I.B.M. 1620

THESE

pour l'obtention du

DOCTORAT de SPECIALITE MATHEMATIQUES (3ème cycle)

Scutenue devant le Jury le 22 février 65

par

Jacques ANDRE

Jury : Monsieur LEGRAS Président

Monsieur DEPAIX

Monsieur PAIR Examinateurs

UNIVERSITE DE NANCY - FACULTE DES SCIENCES

Doyen : M. AUBRY

Assesseur : M. GAY

Doyens honoraires : MM. CORNUBERT - DELSARTE - URION - ROUBAULT -

Professeurs honoraires : MM. CROZE - RAYBAUD - LAFFITE - LERAY - OLY- LAPORTE - EICHHORN - CAPELLE - GODEMENT - DUBREIL - L. SCHWARTZ - IEUDONNE - De MALLEMANN - LONGCHAMBON - LETORT - DODE - GAUTHIER - OUDET - OLMER - CORNUBERT - CHAPELLE - GUERIN - CHEVALLIER - WAHL - ERVE -

Maîtres de conférences honoraires : MM. LIENHART - PIERRET -

PROFESSEURS

MM.			
IRION	Chimie biologique	SCHWARTZ	Exploit. minière
DELSARTE	Analyse supérieure	GAYET	Physiologie
ROUBAULT	Géologie	MANGENOT	Phytopathologie
VEILLET	Biologie animale	MALAPRADE	Chimie
SCHEVIN	Botanique	HADNI	Physique
BARRIOL	Chimie théorique	BONVALET	Mécanique physique
BIZETTE	Physique	KERN	Minéralogie
GUILLIEN	Electronique	BASTICK	Chimie
GIBERT	Chimie physique	DUCHAUFOUR	Pédologie
LEGRAS	Mécanique rationnelle	NEEL	Chimie ind. orga.
BOLFA	Minéralogie	GARNIER	Agronomie
NICLAUSE	Chimie	WEPPE	Minéralogie appli.
FAIVRE	Physique appliquée	BERNARD	Géologie appliquée
AUBRY	Chimie minérale	CHAMPIER	Physique
DUVAL	Chimie	REGNIER	Physico-chimie
COPPENS	Radiogéologie	GAY	Chimie biologique
FRUHLING	Physique	WERNER	Botanique
SUHNER	Physique expérimentale	CONDE	Zoologie
HILLY	Géologie	STEPHAN	Zoologie
LE GOFF	Génie chimique	EYMARD	Cal. Dif. et Int.
CHAPON	Chimie biologique	LEVISALLES	Chimie organique
HEROLD	Chimie industrielle	N.	Physique
		N.	M. M. P.

MAITRES DE CONFERENCES ET PROFESSEURS SANS CHAIRE

MM.			
GOSSE	Génie chimique	Mme HERVE	Math. (propédeutique)
ROCCI	Géologie	AUROUZE	Géologie
VUILLAUME	Psychophysiologie	MARI	Chimie I. S. I. N.
PLAN	Mécanique physique	LAFON	Physique I. S. I. N.
Mme BASTICK	Chimie M. P. C. Epinal	FELDEN	Phy. Théor. & Nucl.
GUDEFIN	Physique	FLECHON	M. P.C.
HORN	Physique	VIGNES	Physique (Mines)
FRENTZ	Biologie animale	Melle HUET	Math. (S. P. C. N.)
		JANOT	M. P. C. Epinal

SECRETAIRE PRINCIPAL : C. CARON

Que Monsieur le Professeur LEGRAS, Directeur du Centre de Calcul Automatique, veuille bien trouver ici l'expression de ma gratitude pour la bienveillante compréhension qu'il a eue à mon égard depuis plus de deux ans.

Monsieur DEPAIX m'honore de sa présence dans mon jury. Je tiens à lui adresser mes remerciements sincères.

J'exprime ma profonde reconnaissance à Monsieur PAIR, initiateur de mes recherches qu'il a toujours suivies de très prés, tant sur le plan théorique que pratique.

Je tiens à remercier tous ceux qui, de quelque façon, ont permis de mener à bien ce travail, et plus particulièrement :

Monsieur PROCIK, Ingénieur IBM à l'Agence de Metz
Mesdames les Secrétaires du Centre de Calcul
La Fondation Nationale des Bourses Zellidja.

Je ne saurais enfin oublier mes amis, Michel CUSEY, Alain FLOC'H et Jean-Marie LAPORTE, qui firent toujours régner, même dans les moments difficiles, la bonne humeur au sein du groupe Algol, et furent pour moi des contradicteurs aimables.

SOMMAIRE

INTRODUCTION

Le travail que nous présen ons ici s'inscrit dans l'ensemble beaucoup plus vaste de la construction d'un compilateur ALGOL entrepris au Centre de Calcul de Nancy sous la direction de Monsieur C.PAIR. Tandis que M.CUSEY créait le programme principal [1] , l'étude des tableaux et celle des procédures étaient respectivement confiées à J.M.LAPORTE et à Madame M.CREHANGE et A.FLOC'H [6] . Notre travail contient deux parties:

- Ecriture de divers sous-programmes:
  - Compilation des nombres
  - Compilation et éxécution des fonctions usuelles
  - Sous-programmes de servitude
- Conception et réalisation des procédures d'Entrée-Sortie

Nos sous-programmes étant appelés par le programme principal ou utilisant certaines zones générales du compilateur, nous allons résumer ici très rapidement certains points décrits par les autres collaborateurs.

Ecriture du programme source [1]

Les symboles de base sont représentés soit par un caractère spécial 1620 [14] , soit par un point suivi d'un de ces caractères, soit par une suite de lettres entre deux points. Ceux que nous aurons l'occasion de citer par la suite sont les suivants:

ALGOL	1620
10	.Z
:	..
;	.,
‘	@
,	$

2)Zones variables à la compilation [1]

La partie du programme ALGOL qui se trouve en mémoire, que nous appellerons parfois"suite d'entrée", est gérée par le compteur MEMSE. Le sous-programme DECAL permet de l'étudier caractère par caractère.

Le compilateur utilise deux piles pour effectuer l'analyse syntaxique:

- La pile O reçoit les équivalents des symboles de base

- La pile V, gérée par le compteur MEMSV, reçoit les adresses des zones affectées aux variables ou expressions, chacune étant suivie d'un indicatif de type.

3) Opérations de compilation [1]

La compilation a pour but de produire des ordres écrits en langage machine qui constitueront le programme objet. La construction de chaque séquence d'ordre est effectuée par une partie du compilateur qui dépend soit du sommet s de la pile O, soit du sommet de la pile O et du caractère ^{x}lu dans la suite d'entrée. Cette partie du compilateur est alors notée $\Sigma(s)$ ou $\Sigma'(s,x)$ et la séquence construite $\Gamma(s)$ ou $\Gamma'(s,x)$.

Deux compteurs, REMAN et MRINT, permettent de gérer le remplissage à l'éxécution des zones affectées aux variables permanentes et aux résultats intermédiaires.

4) Traitement des tableaux [2]

Deux points nous intéressent directement:

- A la compilation, la déclaration d'un tableau produit la création d'ordres machines pour le programme objet qui permettront à l'éxécution de construire un " vecteur de renseignement". Ce vecteur contiendra toutes les informations nécessaires et permettra par divers calculs d'adresses de connaître notamment la dimension du tableau, son type, les valeurs numériques des paires de bornes et les adresses de stockage des composantes.

- L'appel d'une variable indicée produira l'entrée dans un sous-programme permettant de calculer l'adresse réelle de cette composante. Nous n'aurons donc à utiliser ces variables que comme des zones à adresse indirecte.

5) Traitement des procédures [6]

La seule chose dont nous ayons à tenir compte est la suivante: Les ordres du programme objet générés pour un corps de procédure pourront être écrits avec des adressesindéxées. Divers sous-programmes permettront de calculer les adresses effectives. Nous dirons qu'ils dévirtualisent les adresses.

R E M A R Q U E S

On trouvera en annexe les sous-programmes que nous avons écrits(en SPS de [14]). Dans ces listes, ainsi que dans les exemples tabulés que nous serons amenés à donner, la correspondance suivante est à noter:

Tabulatrice	Lire
%	(
¤	)
¢	+
#	=

Le 1620 étant un petit ordinateur, nous avons sacrifié en général la rapidité d'éxécution à un encombrement minimum, sauf pour certains programmes pouvant être très répétitifs.

Chapitre 1

TRAITEMENT DES NOMBRES A LA COMPILATION

Les nombres n'ont pas été traites pas [1] puisqu'ils ne conduisent pas à la création d'ordre machine pour le programme objet. Le sous-programme que nous allons décrire ici (v.Annexe p.1) a pour but de les transformer en constante machine, de mettre l'adresse de cette constante sur le pile V et de produire une carte chargement pour le programme objet. Les deux dernières phases n'offrant aucune difficulté, nous nous étendrons surtout sur le passage d'un nombre ALGOL en constante machine.

1. REMARQUES PRELIMINAIRES

1.1 Définitions

Bien que l'on ait dans [4] la définition:

⟨NOMBRE⟩ ::= ⟨NOMBRE SANS SIGNE⟩ | + ⟨NOMBRE SANS SIGNE⟩ | - ⟨NOMBRE SANS SIGNE⟩

NOMBRE n'intervient dans aucune autre règle de la grammaire d'ALGOL. Les signes + et - pouvant précéder un nombre sans signe sont considérés comme opérateurs additifs et donc traités lors des expressions arithmétiques par le programme principal.

On ne s'occupe donc ici que des nombres sans signe, aussi, aucune équivoque n'étant possible dans ce chapitre, les appellerons nous "nombres".

Par la suite nous utiliserons divers termes dont nous donnons dès à présent les définitions:

- Zéro muet: Tout zéro qui n'est pas précédé d'un chiffre(autre que zéro).
- Partie muette: Une suite de zéros muets.
- Premier chiffre significatif: le premier chiffre (autre que zéro) qui suit éventuellement une partie muette.
- Chiffre significatif: Tout chiffre(y compris zéro) qui suit(ou qui est) le premier chiffre significatif.

1.2 Représentation machine des ENTIERS et des REELS

Les entiers seront des zones de 12 positions de mémoire (" virgule fixe ").

Les réels seront représentés sous forme " flottante" avec une caractéristique de 2 chiffres et une mantisse de 10, sous forme normalisée [14] . La valeur absolue de tout réel devra donc être inférieure à 10^{99}. Si elle devait être inférieure à 10^{-99}, on convient de considérer le nombre comme nul.

1.3 Ecriture ALGOL des nombres

Nous admettons toutes les règles de [4] pour les nombres, en amenant toutefois deux restrictions :

- L'entier qui suit éventuellement le 10 du facteur de cadrage ne devra pas avoir plus de 3 chiffres significatifs.
- Tout nombre du type ENTIER écrit avec plus de 12 chiffres significatifs sera mis en machine sous forme de REEL.

Le sous-programme donnera à un nombre l'un des types ENTIER ou REEL selon la présence d'un point décimal, d'un facteur de cadrage ou d'un nombre de chiffres significatifs supérieur à 12. Tous ces critères ne seront connus qu'au moment où la suite d'entrée sera au niveau du premier symbole de base suivant le nombre. C'est pourquoi les nombres seront initialement construits comme des nombres flottants avec une mantisse de 12 positions, et ensuite seulement mis sous leur forme définitive.

1.4 Construction de la mantisse

Les chiffres significatifs qui apparaissent dans la suite d'entrée seront mis dans le même ordre dans la mantisse, jusqu'à concurrence de 12. N'y entreront pas les zéros muets et les chiffres suivant le douzième chiffre significatif.

1.5 Calcul de la caractéristique

Pour un nombre différent de zéro, la valeur C de la caractéristique du nombre flottant lui correspondant est :

$$C = F + \Pi \qquad (1)$$

où F vaut
- zéro s'il n'existe pas de facteur de cadrage
- la valeur de l'entier qui suit le facteur de cadrage.

Π est le poids de la mantisse:

---

(1) De façon plus générale, on a $C = F + \Pi + A$, où A est un entier dont la valeur dépend de chaque machine. Ici A =0.

Le poids Π de la mantisse est

$$\Pi = \begin{cases} 1 & \text{si le nombre n'est qu'un facteur de cadrage} \\ \sum \omega_i & \text{dans les autres cas, la somme étant étendue à tous les chiffres formant le nombre décimal.} \end{cases}$$

Chaque ω_i est le poids du i-ème chiffre, dont la valeur dépend du type du chiffre et de sa position par rapport au point décimal:

- Zéro muet avant le point décimal $\omega = 0$
  Zéro muet après le point décimal $\omega = -1$
- Chiffre significatif avant le point $\omega = 1$
  Chiffre significatif après le point $\omega = 0$

2 PRINCIPE DE LA TRADUCTION DES NOMBRES

2.1 Syntaxe et sémantique des nombres ALGOL

Le sous-programme principal construit l'arbre de décomposition syntaxique d'un programme ALGOL en considérant les nombres comme des feuilles de cet arbre.

La syntaxe des nombres en ALGOL est écrite tout à fait indépendamment de la sémantique. C'est ainsi qu'elle ne permet pas de tenir compte des poids de chaque chiffre(et plus particulièrement ne donne aucune valeur particuliere aux " zéros muets" suivant le point décimal). Sans rien changer à la généralité d'ALGOL, on peut utiliser pour les nombres une grammaire différente, nettement plus lourde, mais beaucoup plus proche de la sémantique. On étudie donc l'ensemble des nombres comme un langage indépendant du reste d'ALGOL. Il s'agit d'un langage de Kleene [7] dont nous allons donner la grammaire.

2.1.1 K-grammaire des nombres

Afin de ne pas avoir à répéter plusieurs fois des lignes équivalentes, nous condensons l'écriture des règles en utilisant deux symboles, D et C, qui ne font pas partie du vocabulaire. Le premier signifie 0 ou 1 ou 2 .... ou 9; le second 1 ou 2 .... ou 9.

La construction ⟨Partie décimale⟩::=D ⟨Partie décimale⟩ doit alors se lire:

⟨Partie décimale⟩ ::= 0 ⟨partie décimale⟩|
1 ⟨partie décimale⟩|
2 ⟨partie décimale⟩|
3 ⟨partie décimale⟩|
4 ⟨partie décimale⟩|
5 ⟨partie décimale⟩|
6 ⟨partie décimale⟩|
7 ⟨partie décimale⟩|
8 ⟨partie décimale⟩|
9 ⟨partie décimale⟩

n°	Règles de la grammaire
1	⟨Nombre⟩ ::= D|
2	0 ⟨nombre⟩|
3	C ⟨partie significative⟩|
4	. ⟨partie décimale⟩|
5	. ⟨partie décimale forte⟩|
6	10 ⟨entier avec signe⟩|
7	10 ⟨entier sans signe⟩
8	⟨Partie significative⟩ ::= D ⟨partie significative⟩|
9	. ⟨partie décimale⟩|
10	10 ⟨entier avec signe⟩|
11	10 ⟨entier sans signe⟩|
12	D
13	⟨Partie décimale⟩ ::= D ⟨partie décimale⟩|
14	D ⟨fin de partie décimale⟩|
15	D
16	⟨Fin de partie décimale⟩::= 10 ⟨entier avec signe⟩|
17	10 ⟨entier sans signe⟩

18 ⟨Partie décimale forte⟩ ::= 0 |
19 0 ⟨partie décimale forte⟩ |
20 0 ⟨partie décimale⟩ |
21 0 ⟨fin de partie décimale⟩
22 ⟨Entier avec signe⟩ ::= + ⟨entier sans signe⟩ |
23 - ⟨entier sans signe⟩
24 ⟨Entier sans signe⟩ ::= 0 ⟨entier sans signe⟩ |
25 C ⟨entier significatif⟩ |
26 D
27 ⟨Entier significatif⟩ ::= D ⟨entier significatif⟩ |
28 D

2.1.2 Analyse du langage

Pour analyser ce langage, il est inutile d'utiliser une pile: il suffit d'automate fini dont les divers états sont représentés par un jeu de flags. On dispose de quatre positions(F2, F5, F7 et F9), où l'on met ou supprime des flags selon chaque regle. Leur combinaison à un moment donné permet de connaître l'état de l'automate. On aura par exemple les combinaisons suivantes:

Etat	F2	F5	F7	F9
Initial	-		-	
partie significative	-	-	-	
partie décimale		-		
partie décimale forte			-	

A chaque passage d'un état à un autre, c'est à dire à l'identification de chaque règle, on effectue certaines opérations liées à la sémantique que nous indiquerons dans le tableau suivant, en même temps que les opérations sur les flags. Nous indiquons par

SF et CF les opérations consistant respectivement à poser et enlever un flag. Les notations " C dans M" et "C dans CC" signifient respectivement "mettre le chiffre C dans la mantisse", et "mettre le chiffre C dans la caractéristique". Et de même pour " D dans M" et "D dans CC". On ne s'occupe pas ici de la question du nombre de chiffres pouvant entrer dans M ou CC.

règle n°	opération	règle n°	opération
1	D dans M; π:=π+1	15	C dans M
2		16	CF F9
3	C dans M; π:=π+1 ; SF F5	17	CF F9
4	CF F2; CF F7	18	π:=π-1
5	CF F2; CF F7	19	π:=π-1
6	CF F2, F5, F7 et F9	20	π:=π-1
7	CF F2, F5, F7 et F9	21	π:=π-1
8	C dans M; π:=π+1 ; SF F5	22	
9	CF F2; CF F7	23	SF F9
10	CF F2,F5, F7 et F9	24	
11	CF F2, F5, F7 et F9	25	C dans CC
12	C dans M; π:=π+1 ; SF F5	26	D dans CC
13	C dans M	27	D dans CC
14	C dans M	28	D dans CC

2.2 Description du programme

2.2.1 Description en Algol linguistique

Il nous semble intéressant de donner un équivalent en Algol du sous-programme de traduction des nombres que nous avons écrit en SPS. Nous ne cherchons pas du tout à faire un essai de description en ALGOL d'un compilateur ALGOL. Ce que nous avons recherché, c'est un équivalent en ALGOL d'un programme écrit en langage machine. C'est pourquoi l'écriture pourra paraître souvent lourde.

Nous avons écrit ce programme en Algol linguistique, extension de l'Algol proposée par [11] . En fait, nous aurions aussi pu l'écrire en Algol, à condition de supposer que le programme à compiler a déja été transcrit de façon que les symboles de base aient un équivalent numérique.

Nous supposons que le nombre à traduire doit se représenter dans une zone soit sous forme entière, soit sous forme flottante(mantisse de 10 chiffres, suivie des deux chiffres de la caractéristique. On ne tient pas compte ici des questions de flags).

On décrit auparavant quelques procédures, inspirées de sous-programmes utilisés dans le compilateur.

```
début commentaire X représente un entier qui sert dans certaines déclarations de bornes, dont la valeur exacte n'offre aucun intérêt ici.

entier H; chaîne CHIFFRE [10] , SYMBOLE [X] , SE [1] , PILE [X] ;
entier procédure NUM(A); chaine A; commentaire Cette procédure, supposée être standard, est une fonction qui donne à l'entier NUM l'équivalent numérique de la chaine A formée uniquement de chiffres alphanumériques. On suppose également que des zéros sont éventuellement placés à gauche du nombre ainsi formé;

booléen procédure APP(ELT, TABLE); chaine ELT, TABLE; commentaire Cette procédure prend la valeur vrai si l'élément ELT, chaine à un caractère, appartient à la table TABLE;
début entier H;
     pour H := 1 pas 1 jusqua RANG(TABLE)-1 faire si SE = ELEMENT(TABLE,H)
     alors allerà FIN1; APP:= faux; allera FIN;
     FIN1 : APP:= vrai;
     FIN :
fin;

procedure ER(N); entier N; commentaire Cette procédure interrompt la compilation et signale une erreur numéro N;
```



```
procédure DECAL; commentaire Cette procédure est supposée faire avancer la suite d'entrée PILE et affecter à SE le caractère ELEMENT(PILE,X), X étant le niveau de la pile;

commentaire On donne à présent les déclarations des identificateurs que nous utiliserons dans notre sous-programme;
booléen F2,F5,F6,F7,F9; entier M,CC,PI,DEB;
chaine MANT [12] , CARA [3] ;
aiguillage DEBU := KNUL,PTD,EXP;

CHIFFRE := "0" suivide "1" suivide "2" ..... suivide"8" suivide "9";
commentaire de même on suppose construire la table SYMBOLE avec tous les symboles de base de l'Algol scientifique à l'exception du point décimal et de 10;

commentaire L'entrée dans notre sous programme se fait de la façon suivante, que l'on indique pour montrer le contenu de SE;

DECAL; si APP(SE,CHIFFRE) alors allera DEBU1 sinon si SE = "." alors allera DEBU2 sinon si SE= "10" alors allera DEBU3;

DEBU1 : F2:= vrai; DEB:=1;
DEBU11: MANT:= vide; PI:=0; F5:= faux; F6:= F7:= vrai;
DEBU12: CARA:= vide; allera DEBU[DEB];
DEBU2 : DEB:= 2; allera DEBU11;
DEBU3 : DEB:= 3; MANT:= "100000000000";PI:= 1; allera DEBU12;

CHF   : DECAL; si ¬APP(SE,CHIFFRE) alors allera PT;
        si ¬F7 alors allera PTD24;
        si ¬F5 alors allera KNUL;
CHF48 : PI:= PI+ 1;
CHF72 : si ¬F6 alors allera CHF; si MANT[12] ≠ '0' alors allera TCH24;
TCH12 : MANT: MANT suivide SE; allera CHF;
KNUL  : si SE= "0" alors allera CHF; F5:= vrai; allera CHF48;
TCH24 : F2:= F6:= faux; allera CHF;
```

```
PT     : si SE≠"." alors allera FACTCADR; si¬F7 alors ER(1);
PTD    : F7:= F2:= faux; DECAL;
PTD24  : si F5 alors allera CH72;
         si SE≠"0" alors début F5:= vrai; allera TCH12 fin;
         PI:= PI-1; allera CHF;

FACTCADR: si SE≠"10" alors allera FINEX;

EXP    : F2:= F5:= F7:= F9:= faux; DECAL; si SE="+" alors allera CHEX;
         si SE  "-" alors allera CHEX12; F9:= vrai;
CHEX   : DECAL;
CHEX12: si¬APP(SE,CHIFFRE) alors allera FINEX; si¬F5 alors allera TDEX20;
         si CARA[3]≠'0' alors ER(3);
TDEX12: CARA:  CARA suivide SE; allera CHEX;
TDEX20: F7:= vrai; si SE="0" alors allera CHEX; F5:= vrai; allera TDEX12;

FINEX  : si¬APP(SE,SYMBOLE) alors ER(2); si F2 alors allera LTR48;
         si MANT[1]='0' alors début CC:= -99; allera CCNB fin;
         CC:= PI + NUM(CARA) x (si F9 alors -1 sinon 1);
         si CC⩾100 alors ER(3);
         M:= NUM(MANT)+ .5 x 10 (CC-9); M:= M-100;
         si CC⟨ -100 alors début CC:= -99; M:= 0 fin
CCNB   : CC:= CC+100 x M; allera PFONB;
LTR48  : CC:= NUM(MANT); allera PFONB;
```

commentaire PFONB est l'étiquette d'un groupe d'instructions où l'on suppose faire les opérations de perforation de ce nombre traduit, et celles de compilation proprement dite (cf § 3 de ce chapitre);

fin

Remarque: Les règles et opérations que nous avons décrites précédemment peuvent ne pas apparaître toujours de façon très nette. En effet, certaines ont été regroupées de façon à gagner de la place. Le flag F6 (ici valeur logique) est introduit pour les tests en rapport avec le nombre de chiffres significatifs déja placés dans MANT.

2.2.2 Exemples

Voyons ce qui se passe pour le nombre 000 001 0203 - :

Le programme principal se branche en DEBU1 quand il a trouvé le premier zéro, qui est alors en SE. PI prend la valeur 0, MANT et CARA sont vides et on va en KNUL. Comme SE contient 0, on est envoyé à CHF qui appelle le caractère suivant (0). F5 étant FAUX, on retourne en KNUL et on boucle ainsi sur les cinq premiers zéros muets.

En CHF, le 1 entre alors en SE; F5 étant toujours FAUX on retourne en KNUL, mais SE ne contenant pas un zéro, F5 devient VRAI et PI prend la valeur 1. En CH72, F6 étant VRAI et le dernier caractère de MANT étant un blanc, on est envoyé en TCH12 où MANT prend la valeur [1].

En CHF, DECAL met 0 dans SE. Mais cette fois, F5 étant VRAI, on va en séquence en CHF48(où PI croît de 1) et 0 entre dans MANT qui devient: [10]. Et ainsi de suite pour les trois chiffres suivants, MANT ayant alors l'allure suivante: [10203].

On est alors en CHF où DECAL donne à SE la valeur -. APP prend donc la valeur FAUX. On passe donc en PT, FACTCADR et on arrive en FINEX où l'on continue en séquence. F2 étant resté VRAI, on saute en LTR48 qui donne à CC la valeur cherchée: CC= 000000010203.

Soit maintenant le nombre .000012345678987654 $_{10}^{18}$;

Le programme principal ayant détecté le point décimal, se branche en DEBU2 d'où l'on va en PTD par l'intermédiaire de DEBU1. F5 étant FAUX et SE contenant un zéro, on va en séquence et PI prend la valeur -1. De CHF on saute à PTD24 où PI décroît de 1. On boucle ainsi deux autres fois, PI valant alors -4.

En CHF, apparaît alors le 1 qui entre en SE, F5 devient VRAI en PTD24. On saute en TCH12 où MANT prend la valeur [1].

On se retrouve en CHF, PTD24 et aussitôt en CHF72 où le 2 entre dans MANT. Et ainsi de suite jusqu'au second 6. Pi n'a pas bougé, et MANT est alors pleine: MANT=123456789876

Le 8 se présente alors en SE. On va, par PTD24, en CHF72. Là, le dernier élément de MANT n'étant pas un blanc, on se branche en TCH24 où F6 prend la valeur FAUX. Dès lors les chiffres suivants font le circuit CHF, PTD24, CHF72, CHF.... sans qu'aucune action se produise.

DECAL met alors le $_{10}$ dans SE. APP répond non, et on arrive en FACTCADR d'où l'on sort en séquence. Après l'appel de DECAL, SE contient 1 et on saute en CHEX12. Les chiffres 1 et 8 entrent dans CARA comme ceux précédants dans MANT, CARA valant alors 18 . On sort de cette boucle en CHEX12(SE contenant alors le point virgule) qui nous renvoie en FINEX où l'on continue en séquence. On fait alors les calculs:

$$CC = -4 + 18 \times 1 = 14$$

$$M = 123456789876 + .5_{10}{}^{3} = 1234567.90276$$

et enfin , on a bien comme voulu: CC = 1234567902.14

Soit enfin le nombre $_{10}{}^{10}$,

Le programme principal se branche en DEBU3 où MANT prend la valeur 100000000000, et PI la valeur 1. Apres passage en EXP, CARA prend comme précédemment la valeur 10 . En FINEX, on continue en séquence, et CC prend la valeur: CC = 1 + 10 x 1 = 11. Et après arrondi(inutile) sur M, on a en définitive: 100000000011 .

2.2.3 Le programme en SPS

Il est très voisin de celui que nous venons donner en Algol linguistique. les étiquettes de ce dernier sont d'ailleurs calquées sur les références SPS, celles du type LTR48 étant en SPS : LTR + 48.

DECAL a été décrit par [1] . On travaille sur les positions numériques des données alphanumériques pour les chiffres. L'équivalent de NUM se place, quant à la partie " précédée de zéros à droite", dans le bloc LTR60. APP se fait par des tests sur les zones alphanumériques des symboles. MANT et CARA sont en fait des zones, le test " vide?" étant un test "=0?" ou un test d'égalité d'adresses suivant les cas. CC et PI définissent une seule zone, et M et CC se suivent dans la constante de C4 de [1] .

Lors du traitement des réels, l'arrondi doit se faire, lorsque la valeur absolue de CC dépasse 90, après translation (sinon il faudrait faire un arrondi avec une caractéristique de 3 chiffres). La détection du point décimal, du $_{10}$ et du symbole de base suivant le nombre se fait de façon plus complexe que celle indiquée, car nous devons tenir compte du fait que le $_{10}$ est représenté en 1620 par .Z et que nombre de symboles de base commencent également par un point. Dès lors, divers tests doivent être ajoutés, et les branchements sont plus nombreux que ceux indiqués.

3 COMPILATION DES NOMBRES TRADUITS

Une fois le nombre traduit sous forme de constante machine, il reste à faire les opérations de compilation à proprement parler: affecter une adresse au nombre, mettre celle ci dans la pile V et perforer une carte permettant de charger le nombre à l'adresse choisie, lors de l'éxécution.

Nous avons décidé de placer les nombres dans la pile des variables remanentes. Le compteur REMAN de [1] nous donne chaque fois la première position libre dans cette pile. C'est celle que nous prendrons pour l'adresse de la constante. A chaque nombre correspondra une nouvelle adresse. Si par exemple un programme ALGOL contient 15 fois le nombre 25138, 15 zones de 12 positions seraient occupées à l'éxécution par la constante 25138.

Or certains nombres interviennent fréquemment dans un même programme. C'est pourquoi nous chargerons systématiquement ceux de la liste ci-dessous. A la compilation, une fois le nombre mis sous forme de constante machine, on regarde s'il s'agit d'une telle constante courante. Si tel est le cas, aucune carte n'est perforée, et on force la pile V à l'adresse de cette constante. Sinon, une carte est perforée, et l'adresse indiquée par le compteur REMAN est mise dans V.

Après avoir baissé éventuellement piles et compteurs, le sous-programme se branche au programme principal en SYMBOL.

Constantes existant à l'éxécution

ENTIERS	REELS
0	0
1	1
2	2
3	0.5
4	0.25
5	0.10
10	0.01
100	

4 ERREURS DETECTEES

On peut détecter lors du traitement des nombres trois types d'erreurs qui seront signalées par le sous-programme IMPER :

ER 1	Erreur de syntaxe dans l'écriture d'un nombre
ER 2	Un nombre est suivi d'une lettre
ER 3	Un nombre est trop grand ($> 10^{99}$)

Il est à remarquer que certaines erreurs ne peuvent être signalées que par le programme principal, et peuvent ainsi conduire à un libellé différent des trois précédents.

C'est ainsi que si un programmeur désirant taper

A:= $12.3_{10}12$;

soit A:= 12.3.Z12., en 1620

oublie le 3 de la partie décimale, notre sous-programme trouvera .. après le 2, qu'il interprètera comme le symbole de base : , et c'est le sous-programme PRIOR de [1] qui signalera une erreur. Celle ci sera une erreur ER 30 (symboles non comparables) à la suite de la comparaison des priorités de : et de :=.

Chapitre n° 2

TRAITEMENT DES FONCTIONS USUELLES

1. DEFINITIONS

Nous utilisons ici l'adjectif " usuelle" à la place de "standard" dont l'introduction en français a été écartée par le Comité Consultatif du Langage Scientifique [8].

Ainsi que le recommande [4] nous avons adopté certains identificateurs pour les fonctions usuelles de l'analyse qui sont mises sous forme de procédures. On peut considérer que ces procédures sont déclarées dans un bloc plus grand que le programme, leurs identificateurs n'étant pas "réservés". C'est à dire que l'on peut écrire un programme du type suivant, où I et D sont respectivement des instructions et des déclarations:

```
début  D;I;
       début D;I;
             SIGNE : I;
       fin
fin
```

Bien entendu, dans le bloc intérieur où l'on utilise l'étiquette SIGNE, il ne sera pas question d'appeler la fonction usuelle SIGNE.

Jusque plus ample informé, les compilateurs construits en France ont adopté les identificateurs LN et ARCTAN pour les fonctions que l'on nomme toujours LOG et ARCTG. Nous les suivons, avec regrets, pour éviter toute erreur.

Liste des identificateurs des fonctions usuelles:

SIN pour le sinus
COS pour le cosinus
ARCTAN pour la détermination principale de la fonction arctangente
EXP pour l'exponentielle
LN pour le logarithme népérien
RAC2 pour la racine carrée
ABS pour la valeur absolue
ENTIER pour la partie entière
SIGNE pour la fonction qui vaut 1 si l'argument est positif
0 s'il est nul
-1 s'il est négatif

Les six premières opèrent indifféremment sur des arguments du type REEL ou ENTIER. Par contre elles fournissent toutes des valeurs du type REEL.

Les trois autres opèrent également sur des arguments du type REEL ou ENTIER. Mais SIGNE et ENTIER donnent des valeurs du type ENTIER tandisque ABS garde le type de son argument, contrairement à [4] .

A ces neuf fonctions, nous en avons ajouté une autre, CLEF, qu'il serait impossible d'écrire sous la forme d'une procédure ALGOL. On peut la définir ainsi:

booléen procédure CLEF(A); réel A;
commentaire Si A=1 ou 2 ou 3 ou 4 et si la clé de même numéro sur le pupitre est en position haute, on affecte à CLEF la valeur vrai. Dans tous les autres cas, on lui donne la valeur faux;

2. METHODE DE LIAISON

Les fonctions usuelles ne faisant pas l'objet de déclarations ne sont pas traitées par [6] mais peuvent être compilées facilement.

Nous avons adopté la méthode suivante:

- Le compilateur génèrera une liaison de la forme

BTM $\overline{\text{FONCT}}$ ADRES

où FONCT sera l'adresse(indirecte) du sous-programme d'éxécution, et ADRES l'adresse où se trouvera l'argument.

- La valeur de la fonction remplacera l'argument dans la zone d'adresse ADRES.

- Les sous-programmes en langage machine ne seront entrés en mémoire que s'ils sont appelés pas le programme objet. Ce sera l'une des fonctions du programme de chargement translatable.

3. COMPILATION DES FONCTIONS USUELLES

L'objet du sous-programme FCTSTD(Annexe p. 5) est donc de:

- générer l'instruction BTM FONCT ADRES
- remplacer,dans la pile V, FONCT pas ADRES
- poser un flag sur une position déterminée qui sera chargée avec le programme objet et donnera l'ordre au programme de chargement translatable d'entrer en mémoire le sous-programme à utiliser.

Le programme principal se branche en FCTSTD lors du $\Sigma(1)$. On trouve alors dans la pile V l'argument et en dessous un équivalent numérique de l'identificateur de fonction standard permettant de calculer un numéro. Chacune des fonctions usuelles est en effet affectée d'un numéro permettant d'orienter les tests, et surtout de calculer facilement chacune des adresses FONCT.

Avant de générer l'ordre de laison, le sous-programme s'assure que l'argument est du type voulu:

- S'il n'est pas du type ENTIER ou REEL, une erreur est signalée.

- Bien que les six premières fonctions usuelles (les fonctions analytiques) admettent des arguments de type ENTIER ou REEL, les sous-programmes d'éxécution travaillent uniquement sur des réels. Si l'argument est du type ENTIER, un ordre d'aller au sous-programme de conversion CV est alors généré.

Enfin, l'argument étant détruit par la valeur de la fonction le programme construit un ordre d'envoi en pile des résultats intermédiaires de l'argument, si celui ci est un identificateur simple.

4 EXECUTION DES FONCTIONS USUELLES

Tous les sous-programmes (Annexe p. 2) sont translatables et basés sur le même principe:

- Branchement au sous-programme TRAD de [6] pour mettre sous forme réelle l'adresse ADRES qui peut être virtualisée lorsque l'appel de fonction usuelle se situe dans un corps de procédure.

- Calculs proprement dits

- Envoi de la fonction à la place de l'argument

Les sous-programmes des six fonctions usuelles SIN, COS, EXP, LOG, RAC2 et ARCTG ont été adaptés de ceux utilisés par FORTRAN II 9 . Les modifications apportées concernent essentiellement l'utilisation de zones fixes(zones de travail ou constantes), la longueur des nombres(qui est fixe ici, au contraire de FORTRAN) et enfin des questions d'entrée et de sortie dans ces sous-programmes (Il n'existe pas d'équivallent de TRAD danz FORTRAN, et ce dernier laisse la valeur de la fonction dans un pseudo-accumulateur).

4.1 Méthodes d'approximation des fonctions analytiques

Les sous-programmes de [9] utilisent les approximations de Hastings [10] . En général, les calculs sont faits en virgule fixe, caractéristique et mantisse séparément. Cette dernière est d'abord agrandie sur 12 ou 13 positions suivant la fonction. Après les calculs, elle est normalisée à 10 chiffres.

SIN & COS En fait ces deux fonctions sont traitées dans le même sous-programme qui possède deux entrées. Le cosinus d'un angle exprimé en radians est calculé par son développement en série entière. Pour SIN(E), on calcule COS($\pi/2$ - E). La valeur de E est d'abord ramenée dans l'intervale ($-\pi/2$, $\pi/2$) . Pour les arguments dont l'exposant est inférieur à 3, l'erreur n'excède pas 10^{-10}. Sinon, la précision de la mantisse du résultat décroît quand la valeur de l'exposant augmente.

ARCTAN L'évaluation se fait par le calcul d'une série. Le résultat est donné en radians. Sauf pour les résultats dont l'exposant est inférieur à -2, l'erreur maximale est 10^{-10}.

RAC2 Le calcul utilise la méthode de " l'entier impair". Le dernier chiffre de la mantisse est vrai à une unité près.

LN Le logarithme de la partie fractionnaire d'un argument positif est évalué au moyen d'une série. L'exposant est multiplié par Log 10 et le produit est ajouté au logarithme de la fraction, cette somme étant le logarithme de l'argument. Dans le cas d'un argument compris entre 0,99 et 1,01 les premiers digits de son logartihme seront des zéros et le résultat, après normalisation, n'aura pas 10 chiffres significatifs. Dans les autres cas, le dernier chiffre est vrai à une unité près.

EXP La valeur de EXP(E) est calculée en utilisant un développement en série de 10^E. Pour ABS(E) = 227.955 924 2 il se produit un dépassement de capacité (supérieur si A positif, sinon inférieur). La valeur de A est multipliée par log e et le produit est séparé en deux parties: l'une entière et l'autre fractionnaire. La première devient l'exposant du résultat et la seconde fournira la valeur de la mantisse par le calcul d'une série. Si A est positif, l'erreur maximale sur le résultat sera $5\ 10^{-10}$. Si A est négatif, le dernier chiffre de la mantisse sera vrai à une unité près.

5 ERREURS SIGNALEES LORS DU TRAITEMENT DES FONCTIONS USUELLES

5.1 Erreurs à la compilation

La seule erreur signalée est ER 90 : le paramètre d'une fonction usuelle n'est pas une expression arithmétique.

5.2 Erreurs à l'éxécution

Les diverses erreurs repérables sont signalées après une correction imposée. Le programme se déroule sans discontinuité, mais avec des valeurs pouvant être fausses. Cela permet toutefois d'en vérifier le bon déroulement logique.

Code Erreur	Nature de l'erreur	Résultat imposé
F 1	Perte de toute précision dans SIN ou COS	0
F 2	Argument nul dans LOG	-10^{99}
F 3	Argument négatif dans LOG	LOG(ABS(E))
F 4	Dépassement de capacité supérieur dans EXP	10^{99}
F 5	Dépassement de capacité inférieur dans EXP	0
F 6	Argument négatif dans RAC2	RAC2(ABS(E))

Par ailleurs le sous-programme ENTIER utilisant le sous-programme fixe de conversions CV, lors de son utilisation une erreur A 1 pourra être signalée. De même les erreurs F 4 et F 5 pourront apparaître lors de l'exponentiation (A ↑ B) qui utilise dans certains cas les sous programmes EXP et LOG.

Chapitre 3

SOUS-PROGRAMMES FONCTIONNELS

On décrit ici deux sous-programmes d'éxécution dont les ordres de liaison sont générés lors de la compilation par le programme principal.

1 SOUS-PROGRAMME DE CONVERSION DES NOMBRES

Ce sous-programme (fixe) permet, à l'éxécution, de convertir un nombre du type ENTIER au type REEL ou inversement. Il peut également être utilisé par d'autres sous-programmes d'éxécution.

1.1 Liaison normale

CV étant la référence de ce sous-programme(Annxe p.10) et A celle de l'adresse de l'argument, on entre en CV par

```
BT   CV A    Si l'argument est du type ENTIER
BTFL CV A    S'il est du type REEL
```

La sortie se fait par un BB, l'argument converti se trouvant toujours à l'adresse référencée CC(Ō0600).

1.2 Conversion d'ENTIER en REEL

Après avoir testé que le nombre n'est pas nul (auquel cas un " zéro flottant" est directement envoyé en CC), on compte le nombre Z de zéros précédant le premier chiffre significatif et on affecte à la caractéristique la valeur 12-Z. Les chiffres significatifs sont transférés devant cette caractéristique après qu'on les ait fait suivre de Z zéros. La mantisse finale ne devant avoir que 10 digits, un arrondi est éventuellement fait sur le onzième chiffre puis la mantisse est normalisée à 10, et le signe ajusté.

1.3 Conversion de REEL en ENTIER

Après avoir testé que le nombre est différent de zéro (résultat immédiat) et inférieur en valeur absolue à 10^{12} (erreur A 1), on ajoute 0,5 à la mantisse et on en prend la partie entière: Si C est la valeur de la caractéristique, on garde les C premiers chiffres de la mantisse que l'on fait précéder de 12-C zéros. Le signe est ajusté.

1.4 Liaison spéciale

Ce sous-programme CV pouvant être appelé par d'autres sous-programmes où l'on est déja entré par un ordre du type BT(et 1620 n'admettant pas de chaines de reprises), nous avons dû prévoir une liaison spéciale:

L'entrée se fait alors par un B normal en CV, l'argument ayant été auparavant transféré en CV-1.

La sortie (commandée par une bascule indiquée par l'absence d'un flag supprimé lors de l'appel spécial) se fait par un B à une adresse fournie par cet appel spécial, après avoir rétabli le flag-bascule de façon que la sortie suivante puisse se faire par un BB sans perte de temps.

2 EXPONENTIATION

2.1 Liaison

L'éxécution de l'exponentiation A↑B se fait par notre sous-programme AB(Annexe p.11).

Le ⌜ construit lors du $\Sigma(\uparrow)$ par le programme principal génère à la compilation les ordres de liaison :

- Mouvement des arguments A et B aux adresses A et B
- BTM AB RET

où RET est l'adresse de retour au sous-programme objet.

Le résultat est placé à l'adresse A. Selon [4] son type peut dépendre de la valeur de B, valeur connue seulement lors de l'éxécution. Or [1] doit connaître dès la compilation le type du résultat placé en A. C'est pourquoi il nous a imposé que le résultat de l'exponentiation soit toujours du type REEL.

Mais si les arguments A et B sont du type ENTIER et si B est positif, le résultat sera inférieur à 10^{12}(sinon une erreur sera signalée). Cela permettra par la suite une conversion de ce résultat sous forme d'ENTIER. On reviendra plus bas(§ 2.3) sur cette question.

2.2 Méthodes de calculs

Le sous-programme AB a été écrit de façon à être éxécuté dans un minimum de temps.

Soit A et B les arguments de A↑B . On les fait suivre de la lettre E ou R selon qu'ils sont du type ENTIER ou REEL. Et soit AB le résultat(toujours REEL).

En respectant les règles de [4] et compte-tenu de la convention de [1] que nous venons de rappeler, les calculs se feront selon le schéma de principe suivant:

AE↑BE	BE$>$0		Calcul de AExAE x....AE (BE-1) fois en virgule fixe, puis conversion de AB en REEL.
	BE$=$0	AE$\neq$0	AB := 0.0
		AE$=$0	Indéfini
	BE$<$0	AE$\neq$0	Conversion de AE en REEL(AE := AR); Calcul de ARxARx...xAR (1-BE) fois; puis de son inverse.
		AE$=$0	Indéfini

AR↑BE	BE > 0		Calcul de ARxARx...xAR (BE-1) fois en virgule flottante.
	BE < 0		Comme pour AE↑BE
AE↑BR			Conversion de AE en REEL(AR := AE), puis traitement comme pour AR↑BR
AR↑BR	AR > 0		AB := EXP(BR x LOG(AR)) (1)
	AR = 0	BR > 0	AB := 0.0
		BR < 0	Indéfini
	AR < 0		Indéfini

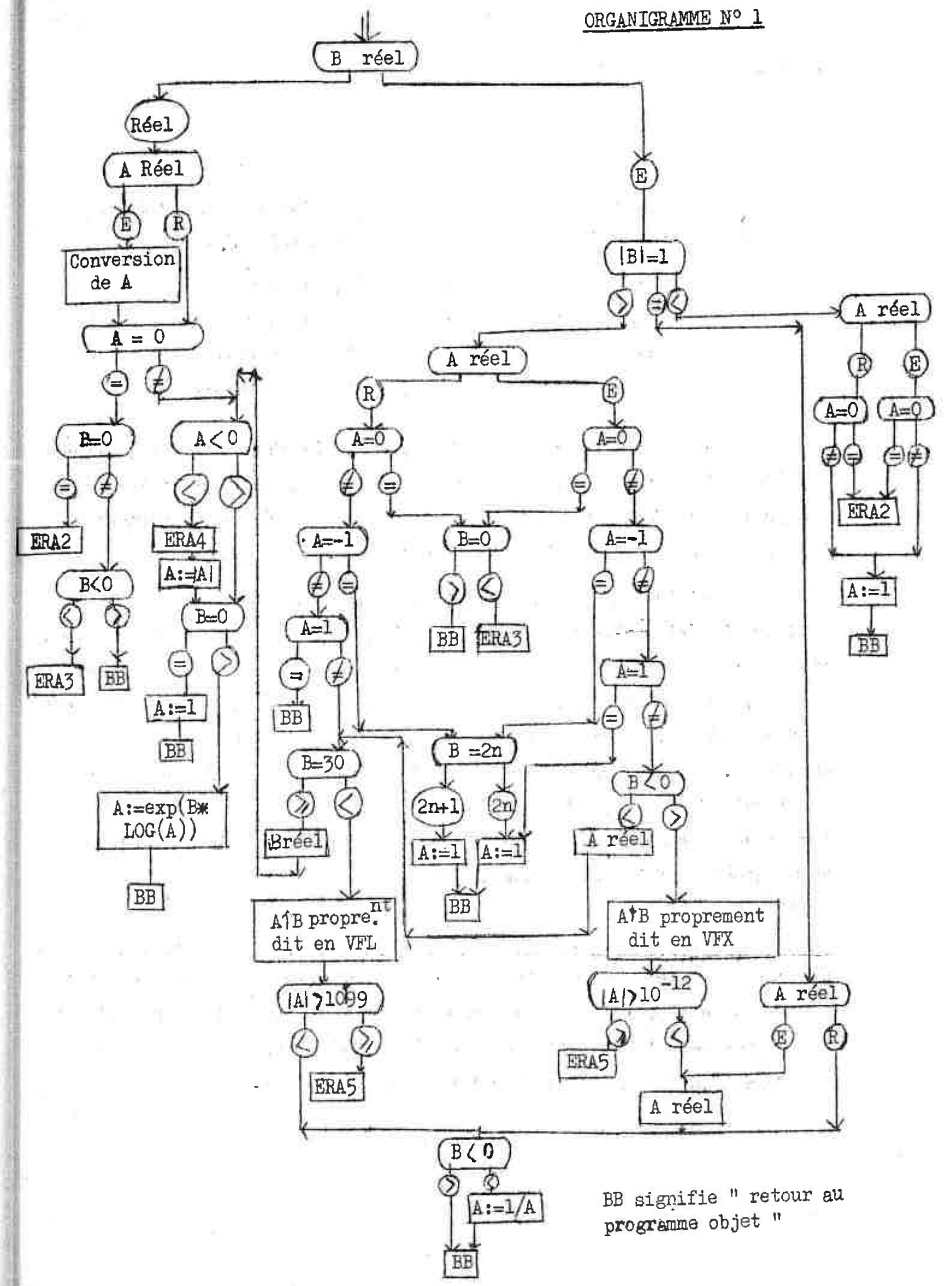
Remarque: Lorsque les arguments sont tous les deux du type ENTIER, la conversion a lieu en des moments différents selon le signe de B. En effet dans le cas où B est positif, on n'admet que des résultats inférieurs à 10^{12} et on gagne du temps à faire les produits en virgule fixe. Par contre lorsque B est négatif, on ne craint pas de dépassements de capacités, et l'on doit alors admettre des résultats inférieurs à 10^{-12}, et en conséquence travailler en virgule flottante.

2.3 Cas particuliers

-i) Il nous a paru intéressant de perdre quelques microsecondes par des tests de façon à éviter dans certains cas des calculs longs et inutiles, le résultat pouvant être connu très rapidement (soit qu'il se trouve déja en A, soit qu'on n'ait qu'à l'y envoyer).

(1) Le programme principal pose lors du ∑(↑)deux flags permettant de charger les fonctions usuelles EXP et LOG quelques soient les types des arguments A et B.

ORGANIGRAMME N° 1



BB signifie " retour au programme objet "

Il s'agit essentiellement des cas suivants, que l'on traite en pratique après avoir éliminé les cas d'indétermination:

AE ou AR = 0	AB := 0 pour tout B
AE ou AR = 1	AB := 1 pour tout B
AE = -1	AB := 1 si BE est pair
	AB := -1 si BE est impair

-ii) Dans le cas de AR↑BE , si ABS(BE)> 30 (1), on se branche au bloc AR↑BR après conversion de BE au type REEL. En effet le calcul par produit de facteurs risque de durer un temps très long, surtout lorsque AR est voisin de 1.0 et BE très grand.

-iii) Nous avons déja signalé que le résultat de AE BE est toujours inférieur(en valeur aboolue) à 10^{12}. En effet il nous est pratiquement impossible de pousser plus loins le calcul en virgule fixe. Mais surtout, nous admettons que si le programmeur a écrit les arguments sous la forme ENTIER, c'est qu'il veut par la suite traiter le résultat comme un ENTIER. Une conversion sera alors faite. Si nous terminions le calcul en virgule flottante, A↑B dépassant 10^{12}, le sous programme CV signalerait une erreur A 1 ; Nous avons jugé préférable de signaler l'erreur comme étant en fait une erreur de dépassement de capacité dans l'exponentiation.

(1) Cette valeur 30 a été fixée approximativement. Il faudrait en fait chercher la valeur correcte par une étude comparée des temps d'éxécution par l'une ou l'autre des méthodes(virgule flottante, ou fonctions usuelles) en fonction des valeurs de B.

L'organigramme n° 1 détaille la façon dont les tests se font.

L'adresse RET pouvant être virtualisée si l'appel de A↑B se fait dans un corps de procédure, le retour au programme objet se fait toujours par le biais du sous-programme INDAX de [6]

2.4 Erreurs signalées par les sous-programmes fonctionnels

Lors de l'emploi de ces sous-programmes, certaines erreurs peuvent être décelées. De même que dans le cas des fonctions usuelles, elles sont signalées, mais ne provoquent pas d'arrêt de l'éxécution du programme objet; un résultat est imposé.

Code erreur	Nature de l'erreur	Résultat imposé
A 1	Un REEL est trop grand en valeur absolue pour être converti en ENTIER	$\pm(10^{12}-1)$
A 2	Les deux arguments de l'exponentiation sont nuls	1
A 3	Exponentiation du type 0↑(-i)	0
A 4	Exponentiation du type A↑BR (où A est négatif et BR un réel)	ABS(A)↑BR
A 5	Dépassement de capacité dans l'exponentiation	10^{99}

Remarques: Le sous-programme d'exponentiation faisant parfois appel aux sous programmes EXP et LOG, une des erreurs F 4 ou F 5 peut être signalée.

Réciproquement, le sous-programme de conversion pouvant être utilisé par d'autres sous-programmes, par exemple par [2] lors du calcul des indices d'une variable indicée, ou au cours du traitement de la fonction usuelle ENTIER, l'erreur A 1 peut apparaître en dehors de son contexte normal.

Chapitre 4

GENERALITES SUR LES PROCEDURES D'ENTREE-SORTIE

Tel qu'il a été défini par [4] , le langage ALGOL, considérant qu'elles dépendent du matériel utilisé, n'aborde pratiquement pas les questions d'entrée-sortie. Mais à mesure qu'apparaissent des compilateurs, ceux ci sont dotés de possibilités d'entrée-sortie de plus en plus perfectionnées, et l'expérience montre qu'elles peuvent être décrites sans incompatibilité avec le langage ALGOL et, c'est ce qui est important, sans aucune référence au matériel utilisé. C'est pourquoi diverses conventions ont été proposées, parmi lesquelles celles du Comité Algol de l'Association for Computing Machinery [12] dont nous nous inspirés tres largement (par la suite nous écrirons " le rapport ACM" chaque fois que nous voudrons faire allusion à ce texte). Entre temps, le Comité International ALGOL définissait(meeting IFIP de Delft, septembre 1963) cinq procédures: in real, out real, in integer, out integer et out string. Maisle rapport n'étant paru que récemment [13] , nous n'avons pu les introduire parmi nos procédures.D'ailleurs ces cinq procédures ne sont que des cas particuliers des notres.

1. Caractéristiques générales

1.1 Les quatre procédures

Nous avons cherché dans la mise au point de ces procédures d'entrée-sortie à concilier au maximum facilité d'écriture et généralité d'emploi. Chacune sera définie comme une procédure (au sens ALGOL), dont la liste des paramètres contiendra toutes les informations nécessaires (liste des variables, formats, opérations d'alignement,etc...) , et dont l'identificateur précisera le sens de l'opération.

La procédure ENTRER permet d'introduire en mémoire une séquence quelconque de variables ou tableaux. Au niveau externe, les données seront écrites en ALGOL, sans que leur représentation ait à être déclarée dans un format, et pourront être séparées par des commentaires.

La procédure SORTIR réalisera à la fois les opérations de composition et d'impression. On pourra sortir en séquence une liste quelconque d'expressions arithmétiques ou booléennes, avec possibilités d'insertions alphanumériques. La représentation externe des données internes pourra être spécifiée dans un format qui ser a soit statique, soit dynamique (c'est à dire dépendant de valeurs connues seulement lors de l'éxécution). Chaque appel de cette procédure ne conduira pas systématiquement à l'impression d'une ligne, ce qui permettra de nombreuses applications non spécifiquement numériques.

Le traitement des formats dynamiques sera possible grâce à une extension d'Algol: la procédure FORMAT qui est en fait une fonction à valeur de chaine.

La procédure SORTAB est un cas particulier de celle SORTIR: elle sera utilisée pour la sortie globale d'un tableau.

Pour permettre la sortie de titres dépendant des problemes traités, nous avons prévu que les procédures ENTRER et SORTIR puissent respectivement entrer puis sortir une chaine alphanumérique (mais sans possibilité de l'utiliser autrement dans le programme).

1.2 Remarques sur la déclaration de ces procédures

Les quatre procédures ENTRER, SORTIR, SORTAB et FORMAT doivent être considérées comme étant déclarées dans un bloc plus grand que le programme lui même. De même que pour les fonctions usuelles (chapitre 3), ces identificateurs suivent les règles Algol relatives aux portées des déclarations.

Le corps de ces procédures est écrit en langage machine: nous pouvons, sans enfeindre la grammaire Algol, admettre des appels dont le nombre de paramètres effectifs soit quelconque.

2. Les organes d'entrée-sortie

Le matériel dont nous disposons comprend deux organes d'entrée-sortie:la machine à écrire de l'unité centrale 1620, et le lecteur perforateur de cartes 1622. Chaque organe pouvant être utilisé dans l'un ou l'autre des deux sens(du niveau interne au niveau externe, ou inversement), nous disposons ainsi de quatre canaux.

2.1 Le lecteur-perforateur de cartes

Voir les détails technologiques dans [14] . Il convient seulement de signaler ici que cette unité permet le traitement des cartes 80 colonnes.

2.2 La machine à écrire

Lorsque la machine à écrire du pupitre est sélectée, elle permet l'entrée d'un nombre quelconque de caractères. Ceux ci sont placés en séquence en mémoire, et détruisent l'état précédent. Nous devons donc limiter le nombre de caractères pouvant entrer en une fois, aussi supposerons nous que cette unité permet l'entrée (ou la sortie) de 80 caractères au maximum. S'il y en a moins, on supposera également que des blancs sont entrés pour compléter la ligne à 80. Bien entendu, il est possible d'entrer une file en la coupant en plusieurs lignes de 80 caractères.

Les entrées et sorties étant très lentes sur cet organe, il est recommandé de s'en servir uniquement comme moyen de communication entre le programmeur et le programme. C'est pourquoi nous ne donnerons pratiquement que des exemples d'entrée-sortie sur cartes.

2.3 Sélection des organes

Les trois instruction procédures d'entrée-sortie devront avoir au moins un paramètre effectif. Le premier permettra la sélection de l'organe. Pour ne pas avoir à le répéter, on définit une fois pour toutes:

⟨Organe⟩ ::= ⟨expression arithmétique⟩

Si au moment de l'appel, cette expression arithmétique vaut 1, l'organe sélecté sera la machine à écrire. Dans tous les autres cas, ce sera le lecteur perforateur de cartes. On admet indifféremment des expressions du type ENTIER ou REEL. Dans ce dernier cas, elle serait convertie à l'éxécution en un nombre ENTIER, après arrondi sur la derniere décimale.

Exemple: SORTIR(si CLEF(1) alors 0.99 sinon 2,....)

Si au moment de l'appel à l'éxécution, la clé numéro 1 est en position haute, la sortie se fera sur machine à écrire; sinon elle se fera sur cartes.

3. Définitions

3.1 Niveaux interne et externe

On qualifiera d'interne tout ce qui a trait à la représentation des variables dans la mémoire centrale, et d'externe ce qui est en rapport avec les organes d'entrée-sortie, que ce soient les supports (cartes ou lignes tapées par machine à écrire) ou les zones de travail (où les données sont lues et déoodées, et celles où les lignes sont composées).

3.2 Notion de ligne et de file

Nous appellerons ligne la suite des 80 caractères(blancs inclus) que l'on peut entrer ou sortir globalement (par exemple, contenu d'une carte). Une ligne forme un tout au niveau externe.

Une file sera une suite quelconque de lignes. Le nombre de caractères dans une file sera donc un multiple entier de 80.

On utilisera indifféremment le terme de blanc ou celui d'espace pour indiquer qu'un caractère d'une ligne est le caractère ␣ d'Algol. Il nous arrivera de le représenter par b ou par une case vide.

4. Plan suivi

Dans une première partie nous définirons les procédures incorporées dans le compilateur: d'abord la procédure ENTRER(ch.5) puis celles de sortie(ch.6).

Puis nous indiquerons comment elles sont traitées, tant à la compilation(ch.7) qu'à l'éxécution(ch.8 & 9). Enfin des exemples concrets seront donnés(ch.11).

5. Détection des erreurs

De très nombreuses erreurs pourront être détectées tant à la compilation que lors de l'éxécution. Tandis que les premieres ne permettent pas de poursuivre la compilation, nous nous sommes arrangés pour que pratiquement aucune erreur d'entrée-sortie ne soit un motif d'arrêt de l'éxécution: divers processus d'auto-correction ou des possibilités de corrections en cours d'éxécution par le programmeur ont été incorporées dans nos sous-programmes.

Dans les différentes règles que nous allons énoncer, il nous arrivera de préciser des limites de validité. Chaque fois, il faudra lire " si cette condition n'est pas remplie, on se branchera à un sous-programme d'erreur".

Le traitement des erreurs fera l'objet du chapitre 10.

6. Comparaisons avec le rapport ACM

Bien que l'on se soit inspiré de très près du rapport ACM, il apparaitra certaines différences entre ce rapport et nos conventions, soit qu'il nous ait pas été possible d'admettre certaines choses, soit qu'elles nous aient paru inutiles. Ces différences seront regroupées dans le dernier paragraphe de chaque chapitre.

En ce qui concerne le présent chapitre de généralités, deux différences importantes sont à signaler:

-i) Le rapport ACM utilise, par exemple, la notation suivante pour entrer 5 données:

INPUT 5 (.....)

Or il est absolument inutile de spécifier ainsi dans la partie identificateur le nombre de paramètres effectifs(voir notre méthode de liaison au chapitre 7). C'est pourquoi nous utiliserons toujours l'écriture:

ENTRER(.....)

quelque soit le nombre des données. Et de même bien sûr pour la sortie.

-ii) Le premier paramètre, dans le rapport ACM, indique le canal choisi. Le sens d'emploi étant déja spécifié par l'identificateur de la procédure, il suffira ici d'indiquer le numéro de l'organe désiré. Le compilateur n'admettant pas encore de cartes de contrôle (extérieures au programme Algol), les numéros d'organes sont imposés et non laissés au choix du programmeur.

Chapitre 5

LA PROCEDURE E N T R E R

1. FORME EXTERNE DES DONNEES

1.1 Données numériques

Aux variables du type ENTIER ou REEL devront correspondre des données écrites sous forme de NOMBRE ALGOL, avec signe éventuel.

Le symbole Algol $_{10}$ pourra être écrit indifféremment .Z ou E ce dernier étant admis de façon que l'on puisse entrer des données qui auront été perforées lors d'une exécution précédente, ou par FORTRAN.

Bien que cela ne soit pas en rapport direct avec la forme externe des données, signalons ici que l'on impose aux nombres sans signe les mêmes restrictions que celles énumérées au chapitre 1.

1.2 Données bocléennes

Les valeurs logiques VRAI et FAUX pourront respectivement être mises sous l'une des formes:

I	O
V	F
VRAI	FAUX

1.3 Séparation des données

Chaque donnée, y compris la dernière, devra être suivie de deux blancs au moins.

Entre deux symboles de base dans une même donnée pourra se trouver un blanc au plus.

Exemple: 12b3bbb sera considéré comme une seule donnée
12bb3bb sera considéré comme deux données.

1.4 Insertions

Chaque donnée pourra être précédée de commentaires que nous appellerons insertions. Ils n'entrent jamais en mémoire interne. Une insertion est une suite de caractères, soumise aux seules conditions suivantes qui évitent de la confondre avec la donnée qui la suit:

- Si cette donnée est numérique, le premier caractère ne peut être ni un chiffre, ni un point, ni un opérateur additif. Il peut n'être un E que s'il est suivi d'une lettre.

- Si c'est une donnée booléenne, le premier caractere ne peut être ni un V, ni un F ni l'un des deux chiffres 0 ou 1.

- Dans les deux cas, une insertion doit être suivie de deux blancs.

La présence de deux blancs, ou d'avantage, à l'intérieur d'une insertion revient à considérer celle ci comme étant formée de deux insertions en séquence. Aussi le caractère suivant les blancs doit-il respecter les règles du premier caractère d'une insertion.

Exemple: Considérons les deux lignes ci dessous:

C	A	R	T	E		1	7		.		8		E	N	T	R	E	E		D	E		E	1		E	T			V		
C	A	R	T	E		1	7		.		8		E	N	T	R	E	E		D	E			E	1		E	T		V		

La première pourra servir d'insertion pour une donnée numérique, la seconde pour une donnée booléenne. Mais l'inverse n'est pas possible: E1 serait pris dans la seconde ligne pour le début d'une donnée numérique, et V,dans la premiere, pour une donnée booléenne.

2. SYNTAXE D'UN APPEL D' ENTRER

⟨procédure ENTRER⟩ ::= ENTRER (⟨organe⟩)|
ENTRER (⟨organe⟩ , ⟨liste d'entrée⟩)
⟨liste d'entrée⟩ ::= ⟨élément de liste d'entrée⟩|
⟨élément de liste d'entrée⟩ , ⟨ liste d'entrée⟩
⟨élément de liste d'entrée⟩ ::= ⟨variable⟩|
⟨identificateur de tableau⟩

La liste d'entrée peut donc contenir un nombre arbitraire d'éléments.

3. S E M A N T I Q U E

Le cas où la procédure ENTRER admet un seul parametre fera l'objet du paragraphe 3.3 de ce chapitre. Dans le cas général, il existe une liste d'entrée non vide: le but de la procédure est d'affecter à chacun de ses éléments une ou plusieurs données externes, ces données étant prises en séquence (et, rappelons le, séparées l'une de l'autre par deux blancs au moins).

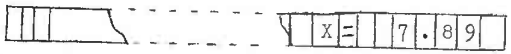
3.1 Opérations d'alignement

Une fois reconnu l'organe d'entrée, la premiere ligne sera appelée, même si éventuellement une ligne précédente n'avait pas été completement lue. Le sous-programme d'éxécution étudiera cette ligne caractère par caractère jusqu'à la reconnaissance des deux blancs qui suivent la dernière des données devant être affectée à un élément de la liste d'entrée. Si ceux ci ne figurent pas sur cette première ligne, de nouvelles lignes seront appellées jusqu'à leur découverte.

A tout appel de la procédure ENTRER doit correspondre une file externe formée de N lignes (N ⩾1). Deux suites de données correspondant à deux appels différents doivent figurer dans deux files disjointes. En revanche, une donnée pourra être écrite " à cheval " sur deux lignes.

Exemple: Soit un premier appel où l'on désire entrer par carte la valeur d'un REEL A, et admettons que l'on fournisse à l'appel de la première ligne, la carte suivante:

Le nombre 123 sera envoyé sous forme réelle à l'adresse deA, et l'éxécution se poursuivra, la carte restant dans le magasin d'alimentation. Supposons que, par quelque bouclage, on revienne à cet appel. Une nouvelle file est introduite, et le contenu de la carte précédente perdu. Si à présent cette file est formée de 5 cartes vierges, 3 cartes ne contenant que des insertions et de la carte suivante:



les huit premières ne feront que " passer " dans l'untié de lecture. La neuvieme ne comporte pas deux blancs après la donnée numérique qui s'y trouve: une dizieme ligne sera appelée. Si la valeur que l'on désire affecter à A est effectivement 7.89 , cette derniere carte devra commencer par une colonne blanche (et le reste sera ignoré).

3.2 Affectation des données

Dans tout ce qui suit, on considère que deux données différentes figurent sur deux lignes disjointes.

Tout se passe comme si le numéro d'organe et les indices des variables indicées étaient d'abord remplacés par leurs valeurs.

-3.2.1 Le numéro d'organe et les indices étant remplacés par des valeurs numériques, l'entrée d'une liste est équivalente à la suite des entrées des éléments de la liste, de sorte que

ENTRER(i,a,b) { a : élément de liste d'entrée ; b : liste d'entrée

équivaut à : ENTRER(i,a); ENTRER(i,b).

-3.2.2 Cas d'une variable simple: l'instruction ENTRER est équivalente à une instruction d'affectation. Dans le cas d'une donnée numérique, le type au niveau interne sera déterminé par le type de l'identificateur et non par celui de la donnée externe. On pourra donc mettre dans la donnée externe d'un ENTIER un point décimal ou un facteur de cadrage; et ceux ci ne sont pas obligatoires pour une donnée de REEL.

Exemple: entier E; réel R; booléen B
ENTRER(2,E, R, B)

éxécuté avec des cartes portant les données -12.787E 2 , 100 et FAUX est équivalent aux affectations:

E := -1279;
R := 10^2 ;
B := faux ;

-3.2.3 Cas d'une variable indicée: Il y a encore équivalence avec une instruction d'affectation, mais il est impossible de lire par un même appel une variable indicée et ses indices. C'est ainsi que le programme suivant:

entier tableau A [1:20, 1: 15] ;
entier I, J; I := 2; J :=3;
ENTRER(2,I,J,A[I,J]);

éxécuté avec des cartes portant les données 4, 5 et 17 serait équivallent aux affectations:

I:=4; J:= 5; A[2,3] := 17;

-3.2.4 Cas d'un identificateur de tableau: les données sont affectées aux composantes du tableau de façon que les deux programmes suivants soient équivalents:

tableau A[B_1:C_1 ,B_2:C_2 , ... ,B_n:C_n];

```
pour I1:= B1 pas 1 jusqua C1 faire
pour I2:= B2 pas 1 jusqua C2 faire
.................
pour In:= Bn pas 1 jusqua Cn faire
   ENTRER(2, A[I1,I2,....,In]);
```

```
ENTRER(2,A);
```

Exemple: entier tableau A[1:3, 1:3] ;
ENTRER(2,A)

éxécuté avec neufs cartes portant les nombres de 1 à 9 , est équivalent à la suite des instructions:

A[1,1]:= 1; A[1,2]:= 2; A[1,3]:= 3;
A[2,1]:= 4; A[2,2]:= 5; A[2,3]:= 6;
A[3,1]:= 7; A[3,2]:= 8; A[3,3]:= 9;

3.3 Entrée de titres

Lorsque la procédure ENTRER n'a qu'un seul paramètre(organe) les opérations suivantes sont effectuées:

- sélection de l'organe de lecture
- appel d'une ligne et d'une seule
- envoi de ses 80 caractères dans une zone appelée PIF.

Cet emploi va de pair avec celui de la procédure SORTIR, écrite elle aussi avec un seul paramètre. Sans trop anticiper sur le chapitre 6, précisons que cette dernière aura pour fonction de:

- sélecter l'organe choisi et faire une opération d'alignement
- envoyer les 80 premiers caracteres de la zone PIF dans la ligne de composition qui sera imprimée ensuite.

La zone PIF étant utilisée à d'autres fins par les procédures SORTIR et SORTAB, cette " reproduction" ne pourra pratiquement être menée à bien que si ces deux appels particuliers de ENTRER et SORTIR sont en séquence.

Exemple d'entrée

Supposons que pour traiter un problème on ait besoin d'entrer une matrice booléenne de dimension m et n, m et n dépendant de chaque problème, et d'une autre matrice 100 x 100 dont tous les éléments soient nuls à l'exception de P d'entre eux, P dépendant également de chaque problème. Le programme pourra s'écrire ainsi:

```
début entier m,n;
      ENTRER(1); SORTIR(2); ENTRER(1,m,n);
      début entier I,J,P; réel X; booléen DC;
            booléen tableau MAT1[1:m,1:n];
            tableau MAT2[1:100,1:100];
            ENTRER(2, MAT1); pour I:=1 pas 1 jusqua 100 faire
                             pour J:=1 pas 1 jusqua 100 faire MAT2[I,J]::0;
            DC::vrai; P::0;
            pour K:=K+1 tantque DC faire
            début ENTRER(2, I , J , X , DC );
                  MAT2[I,J]::X; P::P+1
            fin
            ......
      fin
fin
```

Le programme se déroulera alors de la façon suivante:

-i) Sélection de la machine à écrire. Le programmeur tape par exemple:

PROBLEME M2 ESSAIS DU 25 JANVIER 1965 POUR M(4,15)

Cette ligne sera ensuite perforée sur carte.

-ii) Appel de M et n. Supposons qu'il tape M = 4 N = 15

-iii)Appel de la matrice MAT1: on donne par exemple les cartes:

```
LIGNE 1   V  F  F  F  V  F  F  V  V  F  V  F  F  V  V
LIGNE 2   F  F  V  V  V  V  F  V  V  F  F  F  V  V  F
LIGNE 3   F  V  F  V  F  V  F  F  V  V  V  V  F  F  V
LIGNE 4   V  F  V  F  V  F  F  V  V  V  F  F  F  V  F
```

-iv) Appel des éléments non nuls de MAT2. Soit par exemple

```
M(  17  ,  28  ) =   -12.303 857 E -29          VRAI
M(  55  ,  41  ) =     0 0000 5                 VRAI
M(  89  ,  45  ) =   67.455 678 08              FAUX
```

On aura ainsi P=3, et une matrice MAT2 où tous les éléments, sauf les trois ci dessus, seront nuls.

5. COMPARAISONS AVEC LE RAPPORT ACM

La procédure ENTRER ressemble en première approximation à celle INPUT, avec les réserves suivantes:

5.1 Absence de format

L'utilisation de formats dans une procédure d'entrée est la cause de nombreuses erreurs difficilement corrigibles. Nous les avons donc délaissés. Outre la facilité d'écriture des données externes, on gagne en simplicité pour les sous-programmes d'éxécution.

On perd certainement un peu de temps lors de l'éxécution. On perd surtout de la place à cause des deux blancs devant suivre chaque donnée. Mais ces inconvénients nous ont semblé bien faibles.

5.2 Entrée de données alphanumériques

L'absence de formats interdit l'entrée de données alphanumériques qui puissent être traitées ensuite comme des variables du type ENTIER. C'est pour pallier cet inconvénient que nous avons incorporé la possibilité d'entrée de titres(§ 3.3). Quant aux calculs sur des données alphanumériques, nous jugeons qu'ils relèvent d'avantage d'extensions d'Algol, comme Algol linguistique [11] .

5.3 Séparation des données

La présence de deux blancs apres une donnée revient à affecter au K du rapport ACM la valeur 2. On n'a pas tenu compte des autres délimiteurs, ni des limites P et I, jugeant que l'intérêt gagné par leur emploi serait minime.

5.4 Liste de données

Il ne nous a pas été possible, essentiellement pour des questions d'encombrement, d'introduire la notion de list procedure du rapport ACM. Elle se situe d'ailleurs à un niveau autre que celui seul des entrées-sorties. la même remarque sera faite pour SORTIR.

On n'admet donc que des listes statiques, c'est à dire ayant pour un même appel un nombre fixe d'éléments.

Chapitre 6

LES PROCEDURES DE S O R T I E

1 Généralités sur les sorties

1.1 Approche du problème

Tout comme dans le domaine de l'imprimerie, les opérations de sortie comprennent deux phases principales: la composition d'une ou plusieurs lignes, puis leur impression proprement dite.

- Considérons d'abord le cas élémentaire où un programmeur désire connaître la valeur d'une expression arithmétique. Une première solution serait de donner le résultat tel qu'il se trouve en mémoire. Mais ce serait indigeste et de plus contraire à l'esprit d'Algol, le programmeur n'ayant pas à connaître la structure interne du matériel qu'il utilise. La seconde solution est alors de sortir le résultat sous la forme d'un nombre ALGOL. Et déja apparaissent des difficultés: Si ce résultat vaut 12, doit-on l'écrire sous la forme 12, ou bien 000 12 ou même $.12_{10}2$, toutes trois écritures admises par Algol? On peut évidemment imposer la représentation externe des données, par exemple en sortant les réels comme la succession d'un signe, du point décimal, de dix chiffres significatifs, et du facteur de cadrage avec un signe et deux chiffres. C'est ce qui sera fait ici lors des "sorties normalisées". Mais le programmeur peut désirer une présentation de ses résultats suivant un certain modèle, ou avec un nombre limité de chiffres. Il faut alors qu'il puisse spécifier lors d'un appel de sortie la façon dont il veut composer ses résultats. C'est ce qui se fait en utilisant une chaine(au sens d'Algol), que nous appelons ici chaine-format, dont le contenu écrit dans un langage codé différent d'Algol contient toutes les informations nécessaires.

- Si maintenant ce programmeur désire sortir un très grand nombre de résultats, dont la liste dépende d'instructions conditionnelles, il sera indispensable qu'il sache quelles sont les valeurs qui ont été sorties, c'est à dire qu'il lui faudra la possibilité d'imprimer également des commentaires ou le nom des identificateurs. D'où la nécessité de pouvoir insérer entre des résultats numériques ou booléens des chaines alphanumériques. Cela aussi pourra se faire grace à la chaine-format.

- Supposons enfin que le programmeur veuille sortir neuf résultats de façon qu'ils apparaissent ainsi:

CAS OU X = 17 P1= 1
P2= 38
P3= -427
P4= 0

Tandisque la première ligne contient trois valeurs(17, 1 et 1), les suivantes n'en ont que deux. Il convient donc de pouvoir indiquer à la machine la façon dont les lignes sont remplies et aussi celle dont elles se suivent.

Ce problème des opérations d'alignement conduit à deux types de procédures de sortie selon la manière dont les opérations de composition et d'impression sont traitées. Dans un premier groupe, inspiré de FORTRAN, un appel de procédure de sortie permet la composition puis aussitôt l'impression d'une ou plusieurs lignes. Dans le second groupe au contraire, ces deux phases sont différenciées, une même ligne pouvant être composée par plusieurs appels et imprimée beaucoup plus tard. Cette méthode permettant d'avantage d'applications, c'est celle que nous avons adoptée.

1.2 Caractéristiques générales

L'appel le plus général pour une sortie sera du type:

SORTIR(organe, format, liste)

où "organe" a le sens que nous lui avons déja donné au chapitre 4."Liste" est la liste des expressions que l'on désire sortir. Enfin " format" est une chaine contenant toutes les autres informations nécessaires pour la sortie.

La procédure SORTIR est une procédure de composition qui a pour fonction de remplir progressivement en mémoire une ligne.

Il n'existe pas de procédure d'impression: cette opération se fera soit automatiquement lorsqu'une ligne sera remplie, soit lorsque le programmeur l'imposera par la présence d'une marque d'alignement dans la chaine-format.

2- L E S F O R M A T S

2.1 POSITION DU PROBLÈME

Dans la chaine format seront donc consignées toutes les informations nécessaires à la représentation des résultats. Une telle chaine sera écrite dans un langage spécial dont nous donnerons la grammaire au § 2.2 . Si l'on exclut les délimiteurs(guillemets, virgules...), les marques d'alignement et le caractère N, à chaque caractère de cette chaine sera associé un caractère dans la ligne en cours de composition, immédiatement à la suite de ceux qui y auront été éventuellement placés lors d'un appel précédent.

Les éléments constituant une chaine format se groupent en trois types principaux:

- Formats de nombre ou de variable booléenne: tandisque que ces derniers ne présentent pas de difficultés, les premiers, de par leur généralité, sont complexes. En effet au moment de l'éxécution, il faut connaître le nombre total de postions qui seront occupées, savoir s'il existera un point décimal ou un facteur de cadrage, déterminer les chiffres qui seront envoyés avant le point décimal et ceux qui seront éventuellement remplacés par des blancs ou des zéros,etc... Tout cela sera précisé par une suite de lettres ou de symboles ayant chacun une signification bien précise. A l'éxécution, le nombre épousera rigoureusement la forme de son modèle, chaque " case" de la ligne en cours de composition étant remplie par un caractère dont la nature dépendra de la valeur du nombre et du caractère qui occupe la même place dans le format.

Ainsi, entre le format -BDDD.DDDTBE-ZD et le nombre $-0.12345678_{10}15$, il existera le parallèlisme suivant au niveau externe:

-	B	D	D	D	.	D	D	D	T	B	E	-	Z	D
x	x	x	-	1	2	3	.	4	5	6	7	E	1	2

où nous avons indiqué par un x une case remplie par un appel précédent.

- Formats de titres permettant de placer des commentaires ou des blancs entre ou à l'intérieur des valeurs numériques ou booléennes.

- Des marques d'aligement indiquant que la ligne en cours de composition est terminée et qu'il faut l'imprimer.

Enfin, plusieurs expressions pourront être composées par un même appel; des séparateurs sont alors nécessaires pour éviter des confusions entre deux formats successifs.

La chaine-format sera étudiée concurrement à la liste des paramètres effectifs de la liste de sortie.

2.2 FORMATS DE NOMBRES

2.2.1 Syntaxe

Composants de base:

⟨Multiplicateur⟩ ::= ⟨chiffre⟩ | ⟨chiffre⟩⟨chiffre⟩ | X

⟨Insertion⟩ ::= B | ⟨multiplicateur⟩B | ⟨chaine⟩

⟨séquence d'insertion⟩ ::= ⟨vide⟩ | ⟨séquence d'insertion⟩⟨insertion⟩

⟨Z⟩ ::= Z | ⟨multiplicateur⟩ Z

⟨Partie Z⟩ ::= ⟨Z⟩ | ⟨partie Z⟩⟨Z⟩ | ⟨partie Z⟩ ⟨insertion⟩

⟨D⟩ ::= D | ⟨multiplicateur⟩ D

⟨partie D⟩ ::= ⟨D⟩ | ⟨partie D⟩⟨D⟩ | ⟨partie D⟩ ⟨insertion⟩

⟨partie T⟩ ::= ⟨vide⟩ | T ⟨séquence d'insertion⟩

⟨partie signe⟩ ::= ⟨vide⟩ | ⟨séquence d'insertion⟩ + | ⟨séquence d'insertion⟩ -

⟨partie entier⟩ ::= ⟨partie Z⟩ | ⟨partie D⟩ | ⟨partie Z⟩⟨partie D⟩

Structure des formats:

⟨format d'entier sans signe⟩ ::= ⟨séquence d'insertion⟩ ⟨partie entier⟩

⟨format de partie décimale⟩ ::= . ⟨séquence d'insertion⟩⟨partie D⟩⟨partie T⟩

⟨format de partie exposant⟩ ::= E ⟨partie signe⟩ ⟨format d'entier sans signe⟩

⟨format de nombre décimal⟩ ::= ⟨format d'entier sans signe⟩ ⟨partie T⟩ |
⟨séquence d'insertion⟩ ⟨format de partie décimale⟩ |
⟨format d'entier sans signe⟩ ⟨format de partie décimale⟩

⟨format de nombre⟩ ::= ⟨partie signe⟩ ⟨format de nombre décimal⟩ |
⟨format de nombre décimal⟩ + ⟨séquence d'insertion⟩ |
⟨format de nombre décimal⟩ - ⟨séquence d'insertion⟩ |
⟨partie signe⟩ ⟨format de nombre décimal⟩
⟨format de partie exposant⟩ |
⟨format normalisé⟩

Remarques: i) Cette syntaxe aurait pu être écrite plus simplement, mais ses constructions plutôt gauches ont été ainsi formulées de façon qu'il n'y ait pas d'ambiguités syntaxiques.

ii) ⟨Chaine⟩ correspond à la définition d'Algol

iii) ⟨format normalisé⟩ sera défini plus bas.

2.2.2 Sémantique

2.2.2.1 Multiplicateurs: Un entier sans signe N utilisé comme multiplicateur signifie que le caractère qui le suit est répété N fois. Ainsi 3B est équivalent à BBB. Si N = 0, ce caractère est supprimé. Par exemple BBoZDD = BBDD. Le caractère X utilisé comme multiplicateur indique que la répétition se fait un nombre de fois N qui dépendra de l'appel (voir ci dessous § 3). Lorsque l'on voudra indiquer qu'un multiplicateur est écrit avec le caractere X, on parlera de multiplicateur X.

2.2.2.2 Insertions: La syntaxe du §2.2.1 a été écrite de façon que l'on puisse insérer, n'importe où dans un format de nombre, une chaine délimitée par ses guillemets ouverts et fermés. Le contenu de ces

chaines apparaitra à la sortie au niveau externe à la même place dans le nombre(les guillemets étant supprimés). De même la lettre B peut être placée en tout endroit d'un format de nombre. Au niveau externe apparaitra un blanc.

2.2.2.3 Suppression des zéros et signes: Dans un nombre, à gauche du point décimal, on admet éventuellement un signe, une séquence de Z et une séquence de D (et des insertions).

La convention pour le signe est la suivante:a) s'il n'y a pas de signe dans le format, le nombre doit être positif ou nul.b)Si un signe + est indiqué, le signe, + ou -, apparaitra au niveau externe en accord avec le signe du nombre. c) S'il y a un signe - dans le format, à la sortie on aura un caractère - si le nombre est négatif, un blanc s'il est positif ou nul.

La lettre Z signifie suppression de zéros, et le D impression d'un chiffre, sans suppression de zéro. Chaque D ou Z compte pour un seul digit. Tout chiffre spécifié par un Z sera supprimé, c'est à dire remplacé par un blanc, quand tous les chiffres à sa gauche sont déja des blancs au niveau externe. Un chiffre spécifié dans le format par un D sera toujours imprimé. A noter que le nombre zéro spécifié par un format ne comprenant que des Z donnera au niveau externe une zone de blancs. Un D serait nécessaire pour faire apparaitre au moins le chiffre 0.

Si des suppressions de zéros sont effectuées, le signe (s'il existe) sera imprimé à la place du zéro supprimé le plus à droite.

2.2.2.4 Point décimal: il est indiqué par le caractere"."; il apparait toujours au niveau externe à la même place que dans le format. A sa droite, il est impossible d'indiquer des Z, seuls des D étant admis.

2.2.2.5 Arrondi et troncature: Les réels sortiront au niveau externe apres arrondi sur le chiffre suivant le dernier D. Mais si la lettre T est utilisée dans le format, cet arrondi n'est pas effectué. Arrondi et troncature d'un nombre A dont le format demande d chiffres décimaux sont définis comme suit:

Arrondi 10^{-d} ENTIER(10^{d} A + .5)

Troncature 10^{-d}SIGNE(A) ENTIER(10^{d}ABS(A))

2.2.2.6 Partie exposant: elle est traitée exactement de la même façon que la partie à gauche du point décimal.

2.2.3 Passage de la forme interne à la forme externe des nombres

2.2.3.1 Généralités

Suivant l'emploi des codes Z,D,E etc, et la valeur du nombre à sortir, la représentation externe de ce dernier peut commencer par des blancs ou des zéros. Considérons en effet le cas simple où le nombre vaut -12.3 et soit d'abord -DD.D le format lui correspondant. Là, il n'y a aucun problème: le nombre sera représenté sous l'aspect |-|1|2|.|3|

Si maintenant le détail de format est -ZZZD.D, puisque le nombre n'a que deux chiffres significatifs dans sa partie entière, et qu'on en demande 4 au niveau externe, il serait normal de sortir ce nombre sous la forme -0012.3 . Mais le programmeur, pour des questions de lisibilité des résultats, veut que les zéros pouvant précéder un nombre soient supprimés et remplacés par des blancs: c'est ce qu'il demande en utilisant le code Z. Aux deux premiers Z correspondent des zéros: des blancs vont donc apparaitre à leur place au niveau externe. Le troisième par contre verra sa case occupée par le chiffre 1. Mais s'il en était ainsi, le signe ne précéderait pas immédiatement le nombre car on aurait |-| | |1|2|.|3| ; c'est pourquoi nous avons dit que le signe occuperait la place du dernier zéro supprimé. En définitive, la représentation du nombre sera la suivante:

- Z Z Z D . D

| | | |-|1|2|.|3|

On remarquera que le point décimal occupe le même emplacement au niveau externe qu'au niveau interne. C'est ce que nous appelons "aligner le nombre en fonction du point décimal".

2.2.3.2 Notations

Supposons que le nombre à sortir soit du type REEL, et soit CC la valeur de sa caractéristique, c'est à dire le nombre de chiffres significatifs de sa partie entiere. Supposons enfin que le nombre ne soit pas nul (ce cas est étudié au § 2.2.3.5).

Si on néglige la présence d'éventuelles insertions(voir § 2.2.3.6), la forme externe la plus générale d'un nombre est:

$$+\underbrace{ZZ....ZZ}_{L_1}\ \underbrace{DD....DD}_{L_2}\ .\ \underbrace{DD....D}_{m}\ \underbrace{D}_{L_3}\ E+\underbrace{ZZ....ZZ}_{L_4}\ \underbrace{DD....DD}_{L_5}$$

Soit donc

L_1 le nombre de caractères Z à gauche du point décimal
L_2 " " D " "
L_3 " " D à droite "
L_4 " " Z à droite du E
L_5 " " D " "

où chaque L_i doit être positif ou nul. Ecrire $L_3=0$ revient à dire qu'il n'y a pas de partie décimale. De même, $L_4=L_5=0$ signifie absence de facteur de cadrage.

2.2.3.3 Format de nombre décimal

$L_4=L_5=0$.

Après arrondi éventuel, le nombre au niveau externe est aligné en fonction de la position du point décimal(ou s'il n'en existe pas, en fonction du dernier caractère). L_1+L_2 définit le nombre maximum de chiffres à sortir à gauche du point décimal. Suivant les valeurs respectives des L_i et de CC, on a les différents cas suivants:

- $CC > L_1+L_2$: il est impossible de sortir le nombre car on ne peut pas faire tenir CC chiffres dans moins de CC cases. Par exemple 12345.6 ne pourra être sorti avec le format DD.DDDD ; une erreur serait signalée et ce nombre sortirait sous forme normalisée.

- $L_2 < CC \leqslant L_1+L_2$: il apparait L_2-CC blancs à gauche du nombre.

exemples: format: -ZZZD.T $L_1=3$; $L_2=1$ - Z Z Z D . T
nombre: -12.398 CC=2;

		-	1	2	.	3

format: -ZZZZ $L_1=4$; $L_2=0$ - Z Z Z Z
nombre -1234 CC=4;

-	1	2	3	4

- $CC=L_2$: il y aura exactement L_1 blancs et toutes les positions spécifiées par des D seront occupées par des chiffres significatifs.

Exemple: Format -ZZZD.D $L_1=3$; $L_2=1$; - Z Z Z D . D
Nombre -1.28 CC=1;

			-	1	.	3

$-CC < L_2$: il y aura encore L_1 blancs, mais les L_2-CC digits D seront des zéros.

Exemple: Format: -ZDDD.DD $L_1=1$; $L_2=3$ - Z D D D . D D
Nombre: -12.38 CC=2;

	-	0	1	2	.	3	8

- $-L_3 \leqslant CC < 0$: Le nombre étant inférieur à un, tous les D à gauche du point décimal (s'il en existe) seront des zéros. Il en apparaitra L_3-CC à droite.

Exemple: Format: -ZZDDD.DDD $L_1=2$; $L_2=L_3=3$; - Z D D D . D D D
Nombre: -0.0012345 CC=-2;

	-	0	0	0	.	0	0	1

- $CC < -L_3$: le nombre sort comme zéro(avec signe).

Exemple: Format: -ZZZ.DDDD $L_3=4$ - Z Z Z . D D D D
Nombre: -0.00000012 CC=-6;

			-	.	0	0	0	0

- Enfin dans tous les cas, si $L_3 > 10-CC$, les $L_3-10+CC$ chiffres de droite seront remplacés par des blancs.

Exemple: Z D D D D . D D D D D D D D D

0	1	2	3	.	4	5	6	7	8	9	9		

2.2.3.3 Format de nombre avec exposant

Le problème est ici plus complexe. Considérons en effet un exemple: Soit -Z.ZDDDD.DE+D un format et soit $-0.1234567899_{10}6$ le nombre à sortir. On pourrait, par exemple, le sortir sous l'une des formes:

- Z Z Z D D D D . D E + D

			-	1	2	3	4	.	6	E	+	2
-	1	2	3	4	5	6	7	.	9	E	-	1

Il est bien évident que l'on choisira cette seconde représentation qui donne le maximum de chiffres significatifs. On dit que l'on a cadré le nombre à gauche.

Supposons maintenant que ce même nombre soit à sortir avec le format -ZZ2DDDD.DDDDDDDDE+D; la forme externe demande 15 chiffres, alors que la mantisse n'en possède que 10. On peut donc prévoir diverses représentations dont les extrèmes sont les suivantes:

	-	Z	Z	Z	D	D	D	D	.	D	D	D	D	D	D	D	E	D
	-	1	2	3	4	5	6	7	.	8	9	9	0	0	0	0	E	- 1
et				-	0	0	1	2	.	3	4	5	6	7	8	9	9	E + 4

La première, cadrage à gauche, consiste à ajouter des zéros dans les positions de droite. La seconde, cadrage à droite, place les zéros supplémentaires à gauche. C'est cette dernière représentation qui sera adoptée, car elle donne exactement la forme interne du nombre (rien ne permettant de dire que le chiffre qui suit le dernier 9 soit un 0) et respecte la suppression Z des zéros.

La méthode sera donc de mettre d'abord le nombre sous forme décimale avec le maximum de chiffres significatifs, puis d'ajuster ensuite le facteur de cadrage. Deux cas peuvent se présenter:

- $L_1+L_2+L_3 \leqslant 10$: on ne sort pas plus de chiffres qu'il n'en existe . Les chiffres externes sont cadrés à gauche, la mantisse étant traitée comme un nombre décimal dont la caractéristique serait $CC' = L_1+L_2$; le facteur de cadrage est ensuite sorti comme le serait l'entier $CC-(L_1+L_2)$, les valeurs de L_4 et L_5 devenant celles de L_1 et L_2.

Exemple: Format: -2Z2D.4DE ZD $L_1=L_2=2$; $L_3=4$; $L_4=L_5=1$
Nombre: $-0.1\,34567898_{10}26$ $CC=26$;

On sort d'abord, selon le format -2Z2D.4D le nombre $-.1234567898_{10}CC'$, avec $CC' = 2+2 = 4$, soit:

	-	Z	Z	D	D	.	D	D	D	D
	-	1	2	3	4	.	5	6	7	9

On construit ensuite le facteur de cadrage qui sera sorti comme l'entier $CC-(L_1+L_2) = 26 - 4 = 22$ avec le format +ZD , L_1 et L_2 prenant les valeurs de L_4 et L_5, soit 1 et 1. CC sortira donc sous l'aspect: |+|2|2| .

Le nombre au niveau externe sera donc:

	-	Z	Z	D	D	.	D	D	D	D	E	+	Z	D
	-	1	2	3	4	.	5	6	7	9	E	+	2	2

Remarque: Si $L_1+L_2+L_3 \leqslant 10$, il n'y a pas de zéros Z supprimés.

- $L_1+L_2+L_3 > 10$: le nombre de chiffres significatifs demandés au niveau externe est supérieur à celui au niveau interne. Le cadrage se fait alors à droite; la mantisse est traitée comme si elle était affectée de la caractéristique $10-L_3$, et le facteur de cadrage sort comme l'entier $CC = L_3-10$.

Exemple: considérons le même nombre, mais le format -4Z4D.9DE+DD; La mantisse est d'abord affectée de la caractéristique $CC' = 10-9 = 1$. Puis la caractéristique sort comme l'entier $CC = 26-10 = 25$. Le nombre externe est alors:

	-	Z	Z	Z	Z	D	D	D	D	.	D	D	D	D	D	D	D	D	E	+	D	D
					-	0	0	0	1	.	2	3	4	5	6	7	8	9	8	E	+	2	5

2.2.3.5 Cas où le nombre est nul

Tous les chiffres spécifiés par des D sortent comme des zéros. Le nombre est considéré comme positif.

Exemples:

Z Z D	- Z Z Z Z	+ Z Z	- Z D D . D D D E + D
| | 0 |	| | | | |	| | + |	| | 0 | 0 | . | 0 | 0 | 0 | E | + | 0 |

Ce dernier aspect peut ne pas paraître très heureux, mais il a l'avantage de garder l'homogénéité lors de sorties " en colonnes" et de plus supprime des difficultés dans les programmes d'éxécution.

2.2.3.6 Exemples avec insetions

La présence de B ou d'insertions ne changent pas le problème: Tous les caractères (guillemets exclus) sortent à la place qu'ils occupent dans le format.

Exemples: - Z D B D D D . D D D B D D D

	-	1		2	3	4	.	5	6	7		8	9	8

'A =' B + D D

A	=		-	2	7

D D' E M E C A S'

1	3		E	M	E	C	A	S

Soit enfin A un entier ainsi que I,J et K, tels que l'on ait A = 1000 I + 100 J + K. Soit le format 'A('D','D')= 'DD , et supposons que I,J et K valllent respectivement 6, 3 et 98. On aura au niveau externe:

A(6, 3)= 98

2.3 AUTRES FORMATS

2.3.1 Syntaxe

⟨Partie booléenne⟩ ::= L | 4F | FFFF | F

⟨Format booléen⟩ ::= ⟨séquence d'insertion⟩⟨partie booléenne⟩
⟨séquence d'insertion⟩| ⟨format booléen⟩

⟨Format normalisé⟩ ::= ⟨séquence d'insertion⟩ N ⟨séquence d'insertion⟩

⟨Format de titre⟩ ::= ⟨insertion⟩|⟨ format de titre⟩ ⟨ insertion⟩

⟨marque d'alignement⟩ ::= S| / | ⟨multiplicateur⟩ S| ⟨ multiplicateur⟩ /

⟨Détail de format⟩ ::=⟨format de nombre⟩ | ⟨format booléen⟩|
⟨format de titre⟩|⟨marque d'alignement⟩|⟨détail de format⟩

⟨Element de format⟩ ::= ⟨détail de format⟩| ⟨ marque d'alignement⟩|
⟨élément de format⟩⟨marque d'alignement⟩

2.3.2 Sémantique

2.3.2.1 Format booléen: Une quantité booléenne pourra être mise sous forme externe avec l'un des aspects suivants:

format:	L	F	4F=FFFF
VRAI	1	V	VRAI
FAUX	0	F	FAUX

2.3.2.2 Format normalisé: Suivant le type de l'expression lui correspondant dans la liste de sortie, N sera équivallent à l'un des formats:

Expression de type	ENTIER	-11ZD
"	REEL	-.1ODE-DD
"	BOOLEEN	F

2.3.2.3 Format de titre: Tous les formats étudiés jusqu'à présent donnaient une équivalence entre le niveau interne et celui externe pour une variable réelle, entière ou booléenne. Par contre un format de titre ne comprend que des insertions et ne possède donc pas d'équivalent au niveau interne. Ils servent à sortir uniquement une série de commentaires (titres par exemple) définis en insertions.

2.3.2.4 Marques d'alignement: Les caractères S et / dans un élément de format indiquent des opérations d'alignement. En première approximation, / signifie changer de ligne de composition, et S tabulation. Ce dernier n'a aucune valeur lorsque la sortie se fait sur cartes. Il est possible de faire plusieurs opérations d'alignement en séquence. Ainsi, //// commandera l'impression de la ligne en cours de composition , puis de trois lignes blanches. Les alignements seront décrits de façon complète au paragraphe 4.3.4.

2.4 CHAINES - FORMATS

Les éléments de format tels que nous venons de les décrire peuvent se combiner entre eux pour former une chaine-format, à condition de suivre les règles suivantes:

2.4.1 Syntaxe

⟨format primaire⟩ ::= ⟨élément de format⟩|
⟨multiplicateur⟩ (⟨format secondaire⟩)|
(⟨format secondaire⟩)

⟨format secondaire⟩ ::=⟨format primaire⟩|
⟨format secondaire⟩ , ⟨format primaire⟩

⟨chaine-format⟩::= '⟨format secondaire⟩'

2.4.2 Exemples

4F et 'X=' 3Z.2DE-ZD sont des formats primaires
4F , 'X=' 3Z.2DE-ZD est un format secondaire
3(4F , 'X=' 3Z.2DE-ZD) est un format primaire

Exemples de chaines-formats:

'4(15ZD)/,L,//' '/'
"X=" 5D,8('---')/' '.5DE-D,X(2(20B.8DE-D)/,SN)'
"BAISSER LA CLE NUMERO 2"/,S 'N = ''

2.4.3 Sémantique

Une chaine format est en fait une suite d'éléments de format séparés par des virgules et qui seront interprétés de la gauche vers la droite.

La construction ⟨multiplicateur⟩(⟨format secondaire⟩) signifie que la quantité entre parenthèses est répétée " multiplicateur " fois, une virgule étant insérée chaque fois entre deux répétitions. Si N=0, les parenthèses et leur contenu sont supprimés.

Ainsi 2(20B.8DE-D) sera équivalent à 20B.8DE-D,20B.8DE-D et X(2(20B.8DE-D),S) sera interprété, si X vaut 3 à l'appel , comme

20B.8DE-D,20B.8DE-D,S,20B.8DE-D,20B.8DE-D,S,20B.8DE-D,20B.8DE-D,S

La construction '(⟨format secondaire⟩)' est utilisée pour permettre un nombre de répétitions quelconque de la partie entre parenthèses.

2.5 REMARQUES SUR LES FORMATS

2.5.1 Limites

En pratique, on ne peut pas admettre de chaines formats de plus de 200 caractères(y compris guillemets et autres séparateurs). Cette chaine, une fois les multiplicateurs supprimés (c'est a dire après la répétition), ne devra pas dépasser 800 caractères.

Un détail de format ne pourra avoir plus de 80 caractères au niveau externe. C'est à dire qu'il ne pourra jamais sortir sur deux lignes.

2.5.2 Remarques

- Les expressions suivantes: 4'....' et 4N n'ont pas de sens. Il faudrait respectivement écrire 4('....') et 4(N).

- Il n'existe pas en Algol d'expressions caténaires comme dans certaines extensions(comme Algol linguistique de [11]). Les formats ne peuvent donc pas entrer dans des expressions.

Par exemple, l'appel suivant n'est pas valable:

SORTIR(2, si A>B alors '4D' sinon '3B-DD.D' , A)

Par contre, on pourrait écrire:

SORTIR(2, FORMAT('X(4D),X(3B-DD.D)', si A>B alors 1 sinon 0,
si A>B alors 0 sinon 1),A);

2.5.3 Résumé des codes

B	Espace blanc
D	Chiffre, y compris zéro
E	symbole remplaçant le $_{10}$ d'Algol
F	Booléen (V ou F)
L	logique (1 ou 0)
N	Format normalisé
S	Saut machine à écrire (tabulation)
T	Troncature
X	Multiplicateur arbitraire
Z	Suppression des zéros
-	Impression du signe si c'est -
+	" " " (+ ou -)
.	Position du point décimal
/	Changement de ligne
( ) , '	Délimiteurs

Note: le caractère Algol ␣ est ignoré dans un format.

3 LA PROCEDURE F O R M A T

Dans une chaine-format, un multiplicateur peut être spécifié par le caractère X si sa valeur n'est connue qu'au moment de l'appel. Il faut donc utiliser une procédure qui affectera à X sa valeur numérique et transformera la chaine avec des X en une chaine équivalente sans X. Rappelons que dans une même chaine format peut se trouver un nombre quelconque de multiplicateurs.

Pour cela, on utilise une fonction à valeur de chaine, ce qui constitue en fait une extension d'Algol:

FORMAT(chaine, liste) sera une fonction au sens Algol qui ne pourra être appelée que comme second paramètre de la procédure SORTIR ou de celle SORTAB. Le résultat sera une chaine-format qui sera placée dans le zone PIF où elle ne sera utilisable que par l'une des deux procédures SORTIR et SORTAB.

Son appel sera du type

FORMAT(Chaine de format, X_1, X_2, ... , X_N)

où chaque X_i sera une expression arithmétique appelée par valeur. Chaque X de la chaine-format, qui doit en contenir exactement N, sera remplacé par la valeur du X_i ayant même rang.

Si par exemple on avait:

```
SORTIR(n, FORMAT( 'X(XB,XD.DDTXB),X/',A-I,A,I+2,I-1,I),....);
```

et si A et I valent respectivement 3 et 1 au moment de l'appel, cet ensemble serait équivalent au suivant:

```
SORTIR(n,' 2(3B,3D.DDToB),1/',....);
```

soit encore

```
SORTIR(n, 'BBB,DDD.DDT,BBB,DDD.DDT,/' ,....);
```

Les valeurs des expressions arithmétiques A_i devront être comprises entre zéro(inclu) et cent (exclu).

4 LA PROCEDURE S O R T I R

4.1 Caractéristiques générales

Nous allons aborder l'emploi des procédures de sortie à l'aide d'exemples, les regles précises n'étant définies qu'ensuite.

Considérons d'abord le cas où nous voulons sortir, sur cartes, une ligne contenant les valeurs des entiers M et N avec au plus 5 chiffres; puis la valeur de la variable indicée X[M] sous la forme d'un nombre avec signe, avec un seul zéro à gauche du point décimal et indication de l'exposant; et enfin la valeur de COS(T) en utilisant un format de nombre décimal sans partie exposant. Par ailleurs nous voulons séparer chacune des données par 3 espacements. On peut réaliser cela par l'appel

```
SORTIR(2,'2(BBBZZZZD),3B,+D.DDDDDDE+DDD,3B,
          -Z.DDDDBDDDD/',N,M,X[M], COS(T))
```

Le premier parametre(2) indique une sortie sur cartes. Le suivant est la chaine de format qui spécifie l'aspect externe des valeurs des 4 paramètres qui sont écrits comme paramètres suivants. Si N = 500, M = 0, X[0] = 18061579 et T=3.1415926536, on composera la ligne:

```
bbbbb500bbbbbbb0bbb+1.806158E+007bbb-1.0000b0000
```

ligne qui sera ensuite imprimée comme le demande la barre /. Si elle n'avait pas été mise, d'autres nombres pourraient apparaître sur la même ligne lors d'un appel futur de SORTIR.

L'appel de SORTIR étant un appel de composition, nous obtiendrions la même ligne par les appels suivants:

```
pour I:= N, M faire SORTIR(2, 'BBBZZZZD' ,I);
SORTIR(2,'3B,+D.DDDDDDE+DDD,3B,-Z.DDDDBDDDD',X[M],COS(T));
SORTIR(2,'/');
```

4.2 Syntaxe d'un appel de la procédure SORTIR

⟨Procédure SORTIR⟩ ::= SORTIR (⟨organe⟩) |
SORTIR (⟨organe⟩, ⟨liste de sortie⟩) |
SORTIR (⟨organe⟩, ⟨format⟩) |
SORTIR (⟨organe⟩, ⟨format⟩, ⟨liste de sortie⟩)

⟨Liste de sortie⟩ ::= ⟨expression arithmétique⟩ |
⟨expression arithmétique⟩, ⟨liste de sortie⟩ |
⟨expression booléenne⟩ |
⟨expression booléenne⟩, ⟨liste de sortie⟩

⟨Format⟩ ::= ⟨chaine-format⟩ |
FORMAT(⟨chaine-format⟩, ⟨liste de format⟩)

⟨Liste de format⟩ ::= ⟨expression arithmétique⟩ |
⟨expression arithmétique⟩, ⟨liste de format⟩

4.3 Sémantique

Les cas particuliers d'appels de SORTIR sans format sont étudiés au § 4.4 .

Rappelons que ⟨organe⟩ est une expression arithmétique dont la valeur détermine le numéro de l'organe de sortie à l'éxécution.

4.3.1 Procédure sans liste de sortie

Dans ce cas, le format ne peut contenir que des insertions ou des marques d'alignement. Ceci mis à part, l'éxécution se fait comme pour les appels avec liste.

4.3.2 Liste de sortie

Comme pour ENTRER, on admet une séquence quelconque de paramètres, mais cette fois ces derniers peuvent être des expressions et ne peuvent plus être des identificateurs de tableaux. La liste de sortie sert à spécifier les quantités qui seront mises sous la forme externe selon le format leur correspondant dans la partie format. Ces valeurs sortent dans l'ordre où elles sont écrites.

4.3.3 Parallèle entre la partie format et la liste de sortie

A chaque élément de la liste de sortie doit correspondre un détail de format du même type.

Les éléments de format seront étudiés les uns apres les autres. Ceux ne contenant que des insertions ou des marques d'alignement seront traités immédiatement; les autres produiront l'appel de la valeur de l'expression ayant le même rang dans la liste de sortie.

Exemple: Si E, R et B sont respectivement des expressions de type ENTIER, REEL et BOOLEEN, l'appel

```
SORTIR(n,'3D/','B EST'BFFFF,'R= ','N',E,B,R);
```

sera traité comme les appels successifs:

```
SORTIR(n,'3D/',E);
SORTIR(n,"B EST'BFFFF",B);
SORTIR(n,"R=");
SORTIR(n,'N',R);
```

4.3.4 Opérations d'alignement

Rappelons encore une fois qu'un appel de SORTIR n'impose pas un ordre d'impression.

Une ligne qui a été composée en un ou plusieurs appels, est imprimée lorsque:

-i) Un caractère / est indiqué dans la chaine format. L'impression se fera de façon que le changement de ligne ait lieu à la place occupée par la barre. Dans l'exemple du paragraphe précédent, la valeur de E serait sortie sur une première ligne, et celle de B sur une seconde.

-ii) Un caractère S est indiqué dans le format, la sortie se faisant sur machine à écrire, et aucun taquet de tabulation n'étant posé.

-iii) La totalité des caractères d'un détail de format

ne peut entrer dans la ligne en cours de composition. Cette ligne est imprimée, et une nouvelle ligne est mise en composition, les premiers caractères y entrant étant ceux du détail. Un détail de format sort donc toujours intégralement sur une seule ligne (rappelons que nous avons imposé qu'un tel détail de format doit occuper 80 caractères au maximum au niveau externe).

-iv) Lorsqu'à la fin de l'éxécution une ligne de composition n'a pas été imprimée. Le mot fin est alors équivalent à SORTIR(1,'/'); SORTIR(2,'/').

N caracteres / en séquence produisent l'impression de la ligne en cours de composition, et de N-1 lignes blanches. Lorsque la sortie se fait sur machine à écrire, ligne blanche signifie en fait retour immédiat du charriot.

La tabulation commandée par le caractère S s'effectue de la façon suivante(uniquement si le numéro d'organe vaut 1):

- Impression de la ligne (sauf si elle est vide)
- Saut du charriot jusqu'au premier taquet.

N caractères S en séquence produisent ces deux premières opérations puis N-1 autres sauts.

Remarque. Il existe en fait deux lignes de composition: une pour chaque organe. Elles se remplissent et sont imprimées indépendamment l'une de l'autre.

4.4 Appels particuliers de SORTIR

4.4.1 Sorties normalisées

S'il n'existe pas de format entre le numéro d'organe et la liste de sortie, chaque élément de celle ci sortira sous forme normalisée et sera suivi de deux blancs. Les opérations d'alignement se feront lorsque la condition iii du paragraphe précédent sera remplie. Si A_i est une expression quelconque, l'appel

SORTIR(n, A_1,A_2, ... , A_N)

sera équivalent aux appels suivants:

SORTIR(n,'NBB', A_1);
SORTIR(n,'NBB', A_2);
.....
SORTIR(n,'NBB', A_N);

4.4.2 Sortie de titres

Lorsque l'appel se fait avec un seul paramètre: SORTIR(n), n étant le numéro de l'organe, l'appel est équivalent à la suite des deux ordres: SORTIR(n,'/'); SORTIR(n,"wwww"), où wwww indiquent les 80 caractères de la zone PIF, où ils auront été mis au préalable par un appel de la procédure ENTRER écrite elle aussi avec un seul paramètre(chapitre 5, §3.3). Rappelons que PIF peut être détruite par l'un des appels de SORTIR avec chaine à multiplicateurs, ou de SORTAB.

5 LA PROCEDURE S O R T A B

Cette procédure est utilisée pour la sortie d'un tableau. Son appel est du type:

SORTAB (organe, format, identificateur de tableau)

La partie format(qui peut être une chaine format ou le résultat de la fonction FORMAT), devra contenir exactement deux primaires; le premier: un format de nombre; le second: un format de nombre ou un format booléen suivant le type du tableau.

Exemples d'appels possibles:

```
booléen tableau B[1:10]; entier tableau E[1:5,2:37,-10:-7];
        SORTAB(2,'2D,4F',B)
        SORTAB(2,'DDB',,'A(I,J,K)=ZD.DDE-DD', E)
        SORTAB(2,'2N',B)
```

Exemples d'appels interdits

SORTAB(2,'2D, 3D, FFFF', B) : Il y a trois primaires
SORTAB(2,'2D, 3D', B, E) : Il y a deux identificateurs de tableau.

5.2 Sémantique

Cette procédure a pour but de sortir toutes les composantes en une seule file de plusieurs lignes.

Chaque ligne contiendra dans sa partie droite un certain nombre de composantes du tableau, sorties suivant le second format primaire; dans la partie gauche figureront les indices de la première composante de la ligne, suivant le premier format primaire. Chaque composante sera écrite sur une seule ligne; on ira à la ligne soit lorsqu'il n'y aura plus assez de place pour écrire entièrement une nouvelle composante, soit lorsque l'avant dernier indice du tableau (si le tableau est de dimension deux au moins) changera de valeur. Si le tableau est de dimension un, il y aura par ligne un indice et une seule composante.

Nous définissons plus précisément la procédure SORTAB par un programme de forme Algol:

Tableau A[a_1:b_1 ,a_2:b_2, ... ,a_n:b_n];

SORTAB(c,'Fi,Fe', A), où Fi et Fe sont les deux primaires, sera équivalent à la séquence des instructions:

```
entier Q,P,D,Li,Le,I1,.....,In,J,N;
N:= la dimension du tableau;
Li:= le nombre de caractères externes du format Fi;
Le:= le nombre de caractères externes du format Fe;
Q:=(80-NxLi)÷Le;     D:= bn-an+1;
P:=si D ≤Q alors D sinon Q;
pour I1:= a1 pas 1 jusqua b1 fairé
pour I2:= a2 pas 1 jusqua b2 faire
.....
pour In-1:= an-1pas 1 jusqua bn-1 faire .
```



```
pour In:=an, In+P tantque In≤ bn faire
début SORTIR(c,'/');
    pour J:= I1,I2, ... ,In faire  SORTIR(c,'Fi',J);
    pour J:= 0 pas 1 jusqua si bn-In>Q alors P-1 sinon
    bn-In  faire SORTIR(c,'Fe', A[I1,I2, ...,In-1 ,In+J])
fin
```

La valeur de Q doit évidemment être positive.

Nous allons donner des exemples de telles sorties, d'abord pour un tableau booléen à une dimension, puis de tableau à deux dimensions tel que toutes les composantes ayant le même premier indice sortent sur une même ligne; et enfin d'un tableau à trois dimensions, nécessitant deux lignes pour les variations du dernier indice. Précisons que ce que nous cherchons dans ce programme ce n'est qu'un exemple de sorties de tableau!

```
début entier I,J,K;  Booléen tableau P[1:12];
                     Réel tableau L[1:5,1:5];
                     Entier tableau M[127:128,-2:1,18:22];
booléen procédure PREM(A); entier A; commentaire: la fonction PREM
est affectée de la valeur vrai si A est premier, sinon elle prend
la valeur faux ;
procédure LAGRANGE(A,N); tableau A; entier N; commentaire: cette
procédure calcule la matrice de Lagrange d'ordre N;
SORTIR(2,'/ 4B'NOMBRE  PREMIER'/');
pour I:=1 pas 1 jusqua 12 faire P[I]:=PREM(I);
SORTIR(2,'6BZD,5B4F',P); SORTIR(2,'//');
 LAGRANGE(L,4);
SORTIR(2,"MATRICE DE LAGRANCE D'ORDRE 4'///');
SORTIR(2,''  I  J'4B');
pour J:= 1,2,3,4,5 faire SORTIR(2,''A(I,'D')'5B',J);
```

```
NOMBRE  PREMIER

     1      VRAI
     2      VRAI
     3      VRAI
     4      FAUX
     5      VRAI
     6      FAUX
     7      VRAI
     8      FAUX
     9      FAUX
    10      FAUX
    11      VRAI
    12      FAUX

MATRICE DE LAGRANGE D ORDRE 4

 I  J    A%I,1¤        A%I,2¤        A%I,3¤        A%I,4¤        A%I,5¤
 1  1   1.000000      0.000000      0.000000      0.000000      0.000000
 1  1  -2.083333      4.000000     -3.000000      1.333333     -0.250000
 3  1   1.458333     -4.333333      4.750000     -2.333333      0.458333
 4  1  -0.833333      1.500000     -2.000000      1.166667     -0.250000
 5  1   0.041667     -0.166667      0.250000     -0.166667      0.041667

EXEMPLE 3

 127   -2   18    -127 .002 E 18   -127 .002 E 19   -127 .002 E 20
 127   -2   21    -127 .002 E 21   -127 .002 E 22
 127   -1   18    -127 .001 E 18   -127 .001 E 19   -127 .001 E 20
 127   -1   21    -127 .001 E 21   -127 .001 E 22
 127    0   18     127 .000 E 18    127 .000 E 19    127 .000 E 20
 127    0   21     127 .000 E 21    127 .000 E 22
 127    1   18     127 .001 E 18    127 .001 E 19    127 .001 E 20
 127    1   21     127 .001 E 21    127 .001 E 22
 128   -2   18    -128 .002 E 18   -128 .002 E 19   -128 .002 E 20
 128   -2   21    -128 .002 E 21   -128 .002 E 22
 128   -1   18    -128 .001 E 18   -128 .001 E 19   -128 .001 E 20
 128   -1   21    -128 .001 E 21   -128 .001 E 22
 128    0   18     128 .000 E 18    128 .000 E 19    128 .000 E 20
 128    0   21     128 .000 E 21    128 .000 E 22
 128    1   18     128 .001 E 18    128 .001 E 19    128 .001 E 20
 128    1   21     128 .001 E 21    128 .001 E 22
```

Exemple de sortie de tableaux


```
SORTIR(2,'BBD,BB-D.6D',L);
SORTIR(2,'////'EXEMPLE 3'/');
pour I:= 127, 128 faire
pour J:=-2 pas 1 jusqua 1 faire
pour K:=18 pas 1 jusqua21 faire
      M[I,J,K]:=(si X>0alors 1 sinon -1)x(I+ABS(J)/1000)x10↑(K-1);
SORTIR(2,'-ZZDBB,3B-3DB.3DBB-DD',M);
fin
```

Ce programme donnera les résultats de la page ci contre.

6 COMPARAISONS AVEC LE RAPPORT ACM

Les pages de ce chapitre constituent pratiquement une traduction du rapport ACM puisque nous avons adopté dans sa presque totalité l'ensemble des règles pour les formats et la procédure OUTPUT. Toutefois diverses différences doivent être notées:

6.1 Codes formats

- Ici, un multiplicateur ne doit pas dépasser 100.
- Le caractère C (position d'une virgule entre tranches de chiffres) n'est pas classique en France, donc inutilisé.
- Tous les formats liés aux chaines alphanumériques(formats A,S...) ne peuvent être adaptés pour les mêmes raisons que celles qui nous ont conduits à ne pas admettre l'entrée de données alphanumériques.
- Les formats I,R et L ne présentent guère d'intérêt. Par contre nous ajoutons le format N.
- Ne disposant pas d'imprimante, nous n'avons pas à utiliser de caractère pour le changement de pages. En revanche, nous amenons la possibilité de tabulations.

6.2 Opérations d'alignement

Nous avons pratiquement suivi le rapport ACM, mais le matériel que nous utilisons étant " petit", nous n'avons pas jugé utile d'introduire les notions de contrôle horizontal ou vertical. En conséquence, avec les notations du rapport ACM, nous fixons obligatoirement:

$$L = L' = 1 \quad ; \quad R = R' = \infty$$
$$P = 80 \quad ; \quad P' = \infty$$

De même nous n'avons pas introduit les procédures END DATA, H LIM et CONTROL, leur absence étant facilement compensable par diverses instructions Algol ou indicatifs sur les cartes de données.

6.3 Procédure FORMAT et procédure LIST

Ce sont les principales différences entre le rapport ACM et nos conventions. Ici une liste de sortie ne peut qu'être statique, au sens où nous l'avons précisé pour la procédure ENTRER. Par ailleurs FORMAT est pour nous une chaine fonction, tandis que dans le rapport elle est une instruction procédure. Ces différences primordiales sont essentiellement dues au manque de place, et aussi à un manque de temps.

Une conséquence immédiate de l'absence de procédure LIST est la suivante: La construction

⟨format secondaire⟩ ::= ⟨multiplicateur⟩(⟨format primaire⟩)

n'aura pratiquement de sens que si le multiplicateur est statique. En effet un format du type X(4ZD.DT) laisse entendre que le nombre d'éléments de la liste de format devant sortir suivant ce format est variable, ce que nous n'admettons pas.

6.4 Correction des erreurs

Le rapport ACM recommande que la détection d'une erreur (par exemple un nombre étant trop grand pour le format indiqué) conduise à un programme d'erreur capable de sortir le maximum d'informations sans rien changer au format. Il nous a paru beaucoup plus simple que ce programme d'erreur impose une sortie normalisée , la chaine format n'étant plus prise en considération pour cet appel.

Chapitre 7

COMPILATION DES PROCEDURES D'ENTREE-SORTIE

Les trois instructions procédures ENTRER, SORTIR et SORTAB et la fonction procédure FORMAT présentent cette propriété commune d'être des procédures admettant un nombre quelconque de paramètres effectifs. C'est pourquoi elles sont compilées par le même sous-programme PROCES(voir Annexe).

Le point le plus important est la façon dont on appelle les sous-programmes d'éxécution, et surtout la méthode utilisée pour traiter un nombre quelconque de paramètres. Nous allons donc commencer par étudier le genre de liaison adopté, les opérations de compilations étant décrites ensuite rapidement.

1 LIAISON AUX SOUS-PROGRAMMES

Outre l'ordre de branchement au sous-programme d'éxécution, le compilateur doit fournir à ce dernier:

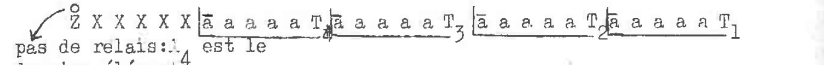
- l'adresse de retour au programme objet
- l'adresse du numéro d'organe et son type
- le nombre de paramètres qui suivent, et pour chacun son type et son adresse.

La liaison peut facilement être générée par l'ordre:

BTM EXECU RETOU

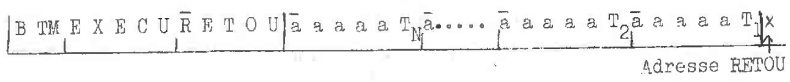
où EXECU sera l'adresse du sous-programme d'éxécution de la procédure, et RETOU celle de retour au programme objet (de même que pour le sous-programme d'exponentiation, on ne peut pas utiliser de chaines de reprises, et on préfère donc réserver l'emploi d'appels par BT aux sous-programmes d'éxédution eux même).

Le reste des informations peut être regroupé dans une table d'adresses: chaque élément de la table comprendra l'adresse A_i du paramètre lui correspondant (et que nous noterons āaaaa quand nous voudrons indiquer qu'elle occupe 5 positions de mémoire), suivie de son type T_i. Toute table aura au moins un élément; les suivants seront placés de la droite vers la gauche. Une marque de mot sur la position de gauche d'une adresse A_i servira de relais et indiquera que A_i fait partie de la table. La fin d'une table de N éléments sera indiquée par l'absence de relais sur ce qui serait la position de gauche de l'élément A_{N+1} s'il existait. Par exemple, un table à 4 éléments aura l'allure suivante(où un x indique un digit quelconque, et z une position de mémoire sans marque de mot):



On appelle sommet de la table la position de T_1. Toutes les informations contenues dans une table pourront être connues par le sous-programme d'éxécution à partir du moment où on lui donne l'adresse du sommet.

Il nous a paru alors particulièrement commode que cette table soit stockée dans le programme objet lui même, entre l'ordre de liaison au sous-programme et l'adresse de retour. Un appel avec N parametres occupera alors les positions suivantes dans le programme objet:



L'adresse RETOU indique en effet à la fois:

- l'adresse de retour au programme objet
- le sommet de la table, plus un.

L'absence systématique de marque de mot sur la position P_6 de l'ordre de liaison(correspondant à la position du U dans cet exemple) indiquera la fin de table.

On remarquera enfin qu'aucune place n'est perdue dans le programme objet, et qu'il y a un seul transfert.

2 COMPILATION DE CES PROCEDURES

Avant de décrire ce que l'on fait, il convient de signaler ce qui a déja été fait par le programme principal.

2.1 Etat du compilateur à l'entrée en PROCES

Considérons par exemple l'appel:

SORTIR(2,'3N',A, COS(X), T[I,J]);

En première approximation, les opérations suivantes sont effectuées par le programme principal:

Une fois reconnu l'identificateur SORTIR, un équivalent numérique est mis dans la pile V, et un indicatif spécial dans celle O. L'adresse machine du nombre 2 est mise à son tour dans V, où elle est suivie d'un indicatif de type. La virgule n'entre nulle part, mais un compteur, CTGOR, progresse de un (on verra par la suite que la compilation de ces procédures est entièrement basée sur le nombre des virgules). Chaque fois que l'on rencontrera une virgule, cette même opération sera faite. A la rencontre des guillemets ouverts, le programme principal se branche à notre sous-programme GUIMET(qui sera décrit au § 3): il a pour but de stocker la chaine& de mettre dans la pile V l'adresse du stockage et un indicatif de chaine. L'adresse de A entre ensuite dans V. Lorsque COS sera reconnu, notre sous-programme FCTSTD (chapitre 2) construira la liaison nécessaire et mettra dans la pile V l'adresse du résultat. De même pour la variable indicée T[I,J],[2] mettant dans V l'adresse de l'adresse (adressage indirect) de cette composante après avoir bati les ordres nécessaires pour son calcul. A la rencontre de la parenthèse fermée, le programme principal entre dans un gamma spécial qui reconnaitra (par le contenu de la pile O) que l'on est dans une procédure d'entrée-sortie et se branchera en PROCES.

La pile V a alors le contenu suivant:

	p.ex.
Adresse de T[I,J] et son type	555551
" du résultat de COS(X)	444440
de A	333333
du stockage de la chaine	222227
du nombre 2	111111
Equivalent de SORTIR	700000

soit, puisque la pile V est construite suivant des adresses décroissantes:

5555514444403333332222271111117000000

On remarquera que les cinq adresses et indicatifs de gauche sont exactement les éléments de la table que l'on veut construire.

2.2 Compilation et construction des ordres

On opère en trois phases:

- Recherche de l'équivalent numérique de l'identificateur de procédure d'entrée-sortie(que nous appelerons de façon plus concise: équivalent)
- Vérifications syntaxiques
- Construction de la liaison.

2.2.1 Recherche de l'équivalent

Le compteur MEMSV indique le sommet de la pile V, alors que l'équivalent se trouve plus bas dans cette pile, à une distance dépendant du nombre de paramètres. Or ce nombre de paramètres est égal au nombre de virgules, plus un. Un calcul d'adresse permet alors de trouver cet équivalent. A noter que si à l'adresse calculée se trouvait autre chose que l'un des quatre équivalents possibles, c'est qu'une erreur aurait été comise lors de l'écriture du programme (par exemple absence de paramètres, ou deux virgules en séquence). Une erreur serait alors signalée et la compilation arrêtée.

Une fois trouvé cet équivalent, diverses initialisations sont faites, et l'adresse EXECU du sous-programme d'éxécution est calculée puis retenue.

2.2.2 Vérifications syntaxiques

Les appels de procédures d'entrée-sortie admettent un nombre quelconque de paramètres effectifs, mais ceux ci doivent suivre certaines règles quant à leur type (par exmple, pour ENTRER, il est interdit qu'un des éléments de la liste d'entrée soit une expression).

Le tableau ci dessous résume les diverses possibilités. On y a employé les notations suivantes:

E Variable de type ENTIER $\bar{E}$ expression de type ENTIER
R " " REEL $\bar{R}$ " " REEL
B " " BOOLEEN $\bar{B}$ " " BOOLEEN
T Identificateur de tableau C Chaine format
V Il peut ne plus exister de paramètres

	Premier Paramètre	second	troisième	suivants
ENTRER	E $\bar{E}$ R $\bar{R}$	E R B T V	E R B T V	E R B T V
SORTIR	E $\bar{E}$ R $\bar{R}$	C E R B $\bar{E}$ $\bar{R}$ $\bar{B}$ V	E R B $\bar{E}$ $\bar{R}$ $\bar{B}$ V	E R B $\bar{E}$ $\bar{R}$ $\bar{B}$ V
SORTAB	E $\bar{E}$ R $\bar{R}$	C	T	//////////////
FORMAT	C	E R $\bar{E}$ $\bar{R}$	E R $\bar{E}$ $\bar{R}$ V	E R $\bar{E}$ $\bar{R}$ V

Tableau des types de parametres admis

Le sous-programme PROCES "remonte" alors la pile V, en vérifiant (par des consultations de tables initialisées lors de la reconnaissance de l'équivalent) que chaque indicatif de type est autorisé. Si tel n'est pas le cas, une indication d'erreur est imprimée et la compilation arrêtée.

2.2.3 Construction de la liaison

Il suffit alors de perforer l'ordre

BTM EXECU RETOU

L'adresse EXECU est déja connue depuis la première phase. L'adresse RETOU se calcule à partir du contenu du compteur CHOB du programme principal (compteur de gestion du programme objet) et du nombre de virgules de l'appel.

Quant à la table de renseignements, nous n'avons pas à la construire: il suffit de perforer le contenu de la pile V compris entre le sommet et l'équivalent. C'est ce qui explique que cette table soit constuite de la droite vers la gauche, puisque c'est une copie de la pile V dont les adresses sont décroissantes.

Une fois perforées les cartes chargement., divers compteurs sont initialisés, et le retour se fait au programme principal en DEBPAR.

2.3 Remarques

2.3.1 Cas d'un appel de FORMAT

- La procédure FORMAT étant toujours appelée à l'intérieur d'un appel de SORTIR ou de SORTAB, diverses précautions doivent être prises lors des calculs utilisant le nombre de virgules.

- Cette procédure est en fait une fonction: il faut remplacer son équivalent dans la pile V par l'adresse du résultat qui sera toujours la même: c'est celle de la zone PIF où le sous-programme d'éxécution de FORMAT laisse toujours la chaine format qu'il a traitée .

2.3.2 Remarques sur l'ordre d'éxécution des instructions générées

Soit par exemple l'appel: SORTIR(A-1, T[I+J,K], SIN(X));

Les instructions seront générées dans l'ordre suivant (qui sera donc aussi celui d'éxécution):

- Calcul de A-1
- Calcul de I+J
- Calcul de l'adresse de la variable indicée
- Calcul de SIN(X)
- Appel de sortir

Les adresses contenues dans la table d'information seront celles des résultats des calculs précédents. C'est ce qui explique que l'on peut considérer que les paramètres et les indices soient " d'abord calculés"(cf chapitre 5,§3.2).

3 STOCKAGE DES CHAINES-FORMATS

Le sous-programme GUIMET(v.Annexe) a pour but de
- stocker les chaines alphanumériques
- mettre l'adresse de cette zone de stockage dans la pile V

3.1 Etude syntaxique de la chaine-format

Lorsque le programme principal détecte des guillemets ouverts, il vérifie que l'on est bien dans une procédure de sortie (ce qui est possible grâce au contenu de la pile 0) puis se branche en GUIMET.

Une chaine ne pouvant qu'être une chaine format, seuls certains caractères sont admis. GUIMET appelle alors, par le sous-programme DECAL, les caractères les uns après les autres et vérifie qu'ils font bien partie de la vingtaine autorisée. En même temps ces caractères sont comptés, de façon à interdire des chaines trop longues, et stockés en attente dans une zone du compilateur (entre les sommets des piles 0 et V). Si dans une telle chaine sont détectés de nouveaux guillemets ouverts, ils doivent correspondre à une insertion:tous les caractères, blancs inclus, sont stockés jusqu'aux guillemets fermés qui doivent délimiter cette insertion.

3.2 Types de chaines

Les chaines-formats peuvent appartenir à l'un des trois types:

- Chaines sans multiplicateurs
- Chaines avec multiplicateurs numériques(statiqas)
- Chaines avec multiplicateurs X (dynamiques)

Pour leur traitement à l'éxécution, il nous est utile de connaitre leur type avant de commencer le décodage: Un indicatif de type est alors placé en tête de chaque chaine.

3.3 Stockage des chaines

La chaine ainsi mise en attente dans le compilateur est ensuite perforée sur une ou plusieurs cartes qui seront chargées avec le programme objet. Lors de l'éxécution, elle se trouvera dans la zone affectée aux réservations permanentes dont la gestion est assurée à la compilation par le compteur REMAN de [1] .

L'adresse de stockage de la chaine, connue grâce à ce compteur REMAN, sera mise dans la pile V, suivie d'un indicatif spécial pour spécifier qu'il s'agit d'une chaine.

Le retour au programme principal se fait en PRIOR.

3.4 Remarque à propos des chaines avec multiplicateurs

Lorsque ceux ci sont numériques, il serait normal que la chaine équivalent soit tout de suite calculée, et soit perforée à la place de l'originale. C'est à dire qu'il faudrait stocker, par exemple,

'BBB,DDDD,N,N' à la place de '3B,4D,2N'

L'intérêt serait d'éviter d'avoir à faire à chaque appel de la procédure de sortie utilisant cette chaine une " démultiplication ".

Nous n'avons pas employé cette méthode pour des raisons de place: le sous-programme de démultiplication est en effet assez volumineux Et de toutes façons, il en faut également un parmi les sous-programmes d'éxécution pour traiter les chaines avec multiplicateurs dynamiques. Nous avons donc préféré perdre un peu de temps à l'éxécution des procédures de sortie(qui ne sont pas des procédures très répétitives, et qui sont de toutes façons lentes), au profit d'un gain de place dans le compilateur et dans le programme objet.

Chapitre 8

EXECUTION DE LA PROCEDURE E N T R E R

Rappelons que l'appel du sous-programme d'éxécution ENTRER se fait par la liaison:

BTM ENTRER RETOU $\underbrace{\bar{a}aaaa}_{A_N}$ T $\underbrace{\bar{a}aaaa}_{A_{N-1}}$ T ..... $\underbrace{\bar{a}aaaa}_{A_2}$ T $\underbrace{\bar{a}aaaa}_{A_1}$ T

où RETOU indique à la fois l'adresse de retour au programme objet et l'adresse (plus un) du sommet de la table d'information. Cette dernière est formée de N ($N \geqslant 1$) éléments comprenant un indicatif T et une adresse $\bar{a}aaaa$ que nous noterons en général A_i pour dire qu'elle est la i-ème de la table.

Le sous-programme ENTRER(voir Annexe) a pour fonction d'entrer N-1 données externes, de les mettre sous forme interne et de les envoyer aux adresses A_2 à A_N suivant les types respectifs indiqués de T_2 à T_N. A_1T_1 sert à spécifier le numéro de l'organe d'entrée.

1 ORGANISATION GENERALE

Ce sous-programme est basé sur l'emploi d'une série de programmes de servitude. Nous verrons dans ce paragraphe l'ordre dans lequel ils peuvent être appelés, et nous en décrirons les principaux au paragraphe suivant.

On ne donne ici que des indications générales, car il n'est pas question d'entrer dans le détail des opérations de gestion. Pour cela on donne un organigramme très grossier qui est détaillé dans le texte. Les sous-programmes auxquels nous ferons allusion auront leur référence écrite en majuscules et mise entre parenthèses.

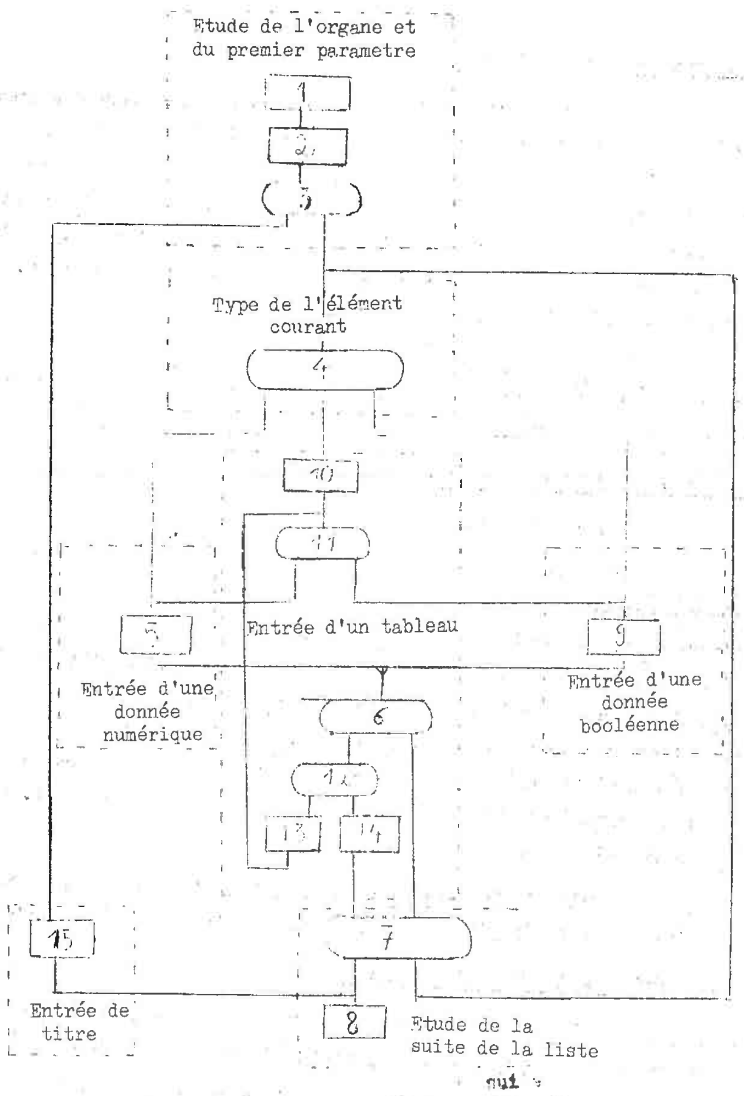
On appelle élément courant l'élément de la table que l'on étudie. A cet élément courant correspond une donnée courante au niveau externe s'il s'agit d'une variable, ou plusieurs données s'il s'agit d'un tableau. On appelle alors donnée courante celle que l'on étudie. L'adresse de l'élément courant sera notée A_i.

Légende de l'organigramme de la page ci-contre

1 - Initialisations
2 - Etude du numéro de l'organe et appel de la première ligne de la file d'entrée (ESTYP)
3 - Si la liste d'entrée est vide, aller à 15
4 - Si l'élément courant est une variable booléenne aller à 9; si c'est un identificateur de tableau, aller à 10
5 - (NB) étudie caractère par caractère (DK) le donnée numérique externe, la met sous forme interne en accord avec T_i, et l'envoie à l'adresse A_i
6 - Si la donnée courante a été affectée à une composante de tableau (dont l'identificateur est l'élément courant) aller à 12
7 - S'il existe un nouvel élément courant (ELSV) aller à 4
8 - Retour au programme objet à l'adresse RETOU
9 - (EBOOL) étudie caractère par caractère(DK) la donnée externe booléenne et l'envoie sous forme interne à l'adresse indiquée en A_i
10- (TAB) étudie le vecteur de renseignement du tableau et calcule notamment l'adresse de la première composante (qu'il envoie en A_i) et de la dernière
11- Si le tableau est de type booléen aller à 9; sinon aller à 5
12- Si la dernière donnée courante a été affectée à la dernière composante du tableau, aller à 14
13- Envoi de l'adresse de la composante suivante du tableau en A_i; aller à 11
14- Rétablir l'adresse A_i modifiée en 10 et en 13; aller à 7
15- Envoyer dans la zone PIF le contenu de la ligne et un indicatif spécial; aller à 8

2 DESCRIPTION DE PROGRAMMES DE SERVITUDE

Plusieurs d'entre eux sont utilisés également lors de l'éxécution de procédures de sortie. Une marque de mot(placée au point 1 du paragraphe précédent) permet de faire certaines opérations uniquement lorsque l'appel de ces programmes de servitude est fait par ENTRER.



Organisation générale de l'éxécution de la procédure ENTRER

2.1 Etude du numéro d'organe : ESTYP

Ce sous-programme est commun aux trois sous-programmes d'éxécution des instructions/procédures d'entrée-sortie. Il calcule, en fonction de l'adresse RETOU l'adresse du sommet de la table. Cette adresse, qui est celle de T_1 est mise dans le compteur SCT, et celle A_1 en SCS. Le nombre contenu à l'adresse A_1 indique le numéro de l'organe. En fonction de son type (précisé par T_1) et de sa valeur par rapport à un, diverses initialisations sont effectuées.

Lorsque ce programme est appelé par ENTRER, le compteur SZS (voir § 2.3) est initialisé de façon qu'au premier appel de DK, la première ligne de la file soit entrée en mémoire.

2.2 Etude de la table d'informations : ELSV

Son appel est du type BTM ELSV NON

Ce sous-programme est utilisé par tous les sous-programmes d'entrée-sortie. Au moment de l'entrée en ELSV, le compteur SCT indique l'adresse T_i du dernier élément courant. ELSV cherche alors la présence d'un relais sur la position de gauche de l'adresse A_{i-1}. S'il en existe, A_{i-1} fait encore partie de la table; les compteurs SCS et SCT sont forcés aux adresses respectives de A_{i-1} et de T_{i-1}, cette dernière ayant auparavant été dévirtualisée par le sous-programme TRAD de [6] au cas o. l'appel d'ENTRER aurait eu lieu dans un corps de procédure ALGOL. Le retour se fait par un BB normal.

S'il n'existe pas relais, la table s'arrête en A_i et tous ses éléments ont déja été étudiés. On sort à l'adresse NON de l'appel.

2.3 Gestion de la file d'entrée : DK

Ce programme sert à effectuer automatiquement les opérations d'alignement et à isoler le caractère qui suit le dernier étudié.

La lecture des lignes se fait dans une zone de 80 positions alphanumériques appelée ZIN. Le compteur SZS indique la position du caractère en cours d'étude.

A chaque appel de DK, on regarde si ce compteur SZS indique l'extrèmité supérieure de la zone ZIN. Auquel cas une nouvelle ligne est entrée dans la zone ZIN, détruisant son contenu, et le compteur SZS réinitialisé. Sinon, ce compteur progresse d'une position alphanumérique.

Un système de bascules permet en outre de supprimer chaque blanc situé entre deux symboles de base.

2.4 Etude des données numériques externes : NB

2.4.1 Comparaison avec la compilation des nombres

Il existe une très grande analogie entre ce programme NB et celui décrit au chapitre 1 pour la compilation des nombres. En effet tous deux ont pour fonction de traduire un nombre écrit en ALGOL en un nombre machine équivalent. Toutefois quelques différences sont à noter:

- La K-grammaire décrite au chapitre 1 n'est pas identique à celle utilisée ici puisqu'on peut entrer un nombre avec signe. Il a donc fallu modifier légèrement les règles et les opérations lors du passage d'un état à l'autre. Cela étant minime, on ne donne pas la grammaire utilisée ici.

- Un nombre peut être précédé de commentaires et la fin est caractérisée par la présence de deux blancs.

- Le type et l'adresse sont imposés par le contenu AT_i et A_i de la table d'information.

2.4.2 Méthode pratique

Apres diverses affectations de compteurs ou zones (qui correspondent à la mise de l'automate à son état initial), le programme DK est appelé jusqu'à la rencontre d'un caractère qui ne soit pas un blanc. Si ce caractère indique le début d'un nombre(c'est à dire si c'est un signe, un point, un E ou un chiffre), on entre à proprement parler dans l'automate. Sinon, on est dans une insertion: DK est appelé jusqu'à la rencontre de deux blancs suivis d'un caractère, et l'étude recommence.

La traduction des nombres se fait, à quelques détails près, de la même façon qu'à la compilation.

Lorsque DK indique la présence de deux blancs, le nombre est

mis sous la forme machine correspondant au type indiqué par T_i et envoyé à l'adresse A_i.

La sortie se fait normalement au point 7 du §1, mais un jeu de bascules permet, lorsque le nombre est en fait affecté à une composante de tableau de retourner au point 12 du § 1 (cf ci dessous, § 2.6)

2.5 Etude des données booléennes: EBOOL

Le programme EBOOL est construit sur le même principe que le précédent. Mais au lieu d'étudier un nombre, il suffit d'analyser les caractères qui doivent composer la donnée externe. Cette étude se fait par des comparaisons. La sortie est commune avec celle de NB.

2.6 Entrée d'un tableau

Ce programme permet d'entrer toutes les composantes d'un tableau. Il faut donc que lorsque la première a été entrée, l'on n'appelle pas le sous-programme ELSV mais quel'on calcule l'adresse de la suivante et qu'on retourne à l'un des programmes EBOOL ou NB selon le type du tableau.Ces diverses opérations se font en plusieurs étapes:

A la découverte d'un tableau dans la table d'information, le programme TAB utilise le vecteur de renseignement construit par 2 (dont l'adresse est celle A_i donnée par la table), calcule à partir de la dimension et des valeurs de paires de bornes les adresses de la première et de la dernière composante de ce tableau. Une bascule est forcée dans le sens indiquant la présence d'un tableau et l'adresse de la première composante est envoyée en SCS. Selon le type T_i du tableau, on va en EBOOL ou en NB.

Au retour d'un de ces programmes, on compare l'adresse contenue en SCS avec celle de la dernière composante. Si elle est inférieure, on augmente SCS du nombre de positions occupées par une composante (qui dépend du type du tableau), et on retourne en EBOOL ou en NB. Si cette composante est la dernière, le tableau est désormais intégralement chargé, la bascule est détruite et on retourne au point 7 du § 1.

Chapitre 9

EXECUTION DES PROCEDURES DE S O R T I E

Le traitement de ces procédures à l'éxécution est basé sur le même principe que celui de la procédure ENTRER: étude de la table d'information. Mais les choses sont beaucoup plus complexes ici puisque l'on doit tenir également compte des informations données par la chaine format quand il en existe. En pratique, ce sont ces dernières qui fixent l'ordre des opérations et commandent l'étude de la table.

1 ORGANISATION GENERALE

Une fois reconnu le numéro d'organe et l'existence d'une chaine format, on étudiera cette dernière primaire par primaire.

1.1 Notations et définitions

Rappelons qu'un primaire peut être mis sous la forme ADF où A indique une séquence de marques d'alignement , D un détail de format (c'est à dire un format de nombre, un format booléen ou un format de titre), et F de nouvelles marques d'alignement. DF constitue un élément de format.

On appellera primaire courant, le primaire que l'on étudie; donnée courante la donnée interne devant sortir avec le format du primaire courant.

1.2 Organigramme général

Dans une première phase, nous allons voir comment sont séparés les cas de sortie sans format et ceux où on utilise un format.

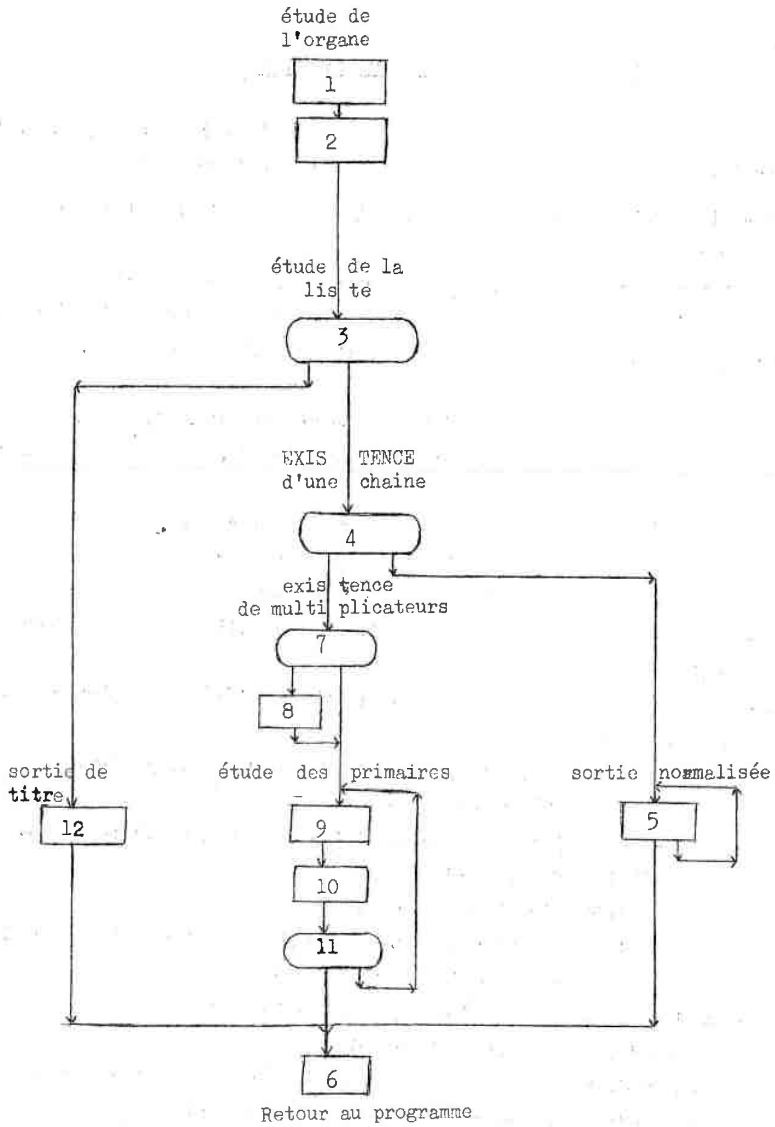
Ensuite, nous étudierons le " décodage " des primaires.

Nous employons la même présentation que pour l'organisation générale de l'éxécution de la procédure ENTRER:

1 - Initialisations
2 - Etude du numéro de l'organe et choix de la ligne de composition(ESTYP)
3 - Si la table d'information ne contient qu'un élément(ELSV) aller à 12
4 - Si l'élément courant est une chaine-format aller à 7
5 - Sortir la donnée courante avec un format normalisé (NORMA) ainsi que toutes les autres de la table d'information (ELSV)
6 - Retour au programme objet
7 - Si la chaine-format ne contient pas de multiplicateurs aller à 9
8 - Remplacer cette chaine par la chaine équivallente (DEMUL)
9 - Traitement direct des alignements A du primaire courant
10 - Etude du détail de format D du primaire courant
- i) Décompte du nombre de positions occupées au niveau externe et éventuellement opérations d'alignement
- ii) Traitement direct des alignements s'il en existe
-iii) Recherche du type de format et composition: si le format est un format de nombre ou un format booléen, la donnée courante est appelée par le biais de la table d'information (ELSV)
- iv) Traitement de la fin F de l'élément de format
11 - Si le primaire courant est le dernier de la chaine aller à 6; sinon étude du primaire suivant en retournant à 9
12 - Changer de ligne (LIGNE) et sortir le commentaire stocké en PIF; aller à 6

1.3 Notion de décodage

On appelle ainsi un ensemble de sous-programmes permettant d'étudier caractère par caractère une chaine format. Chaque caractère commande l'appel d'un sous-programme déterminé par une consultation de table d'adresses. Ces sous-programmes se rebranchent soit à un nouveau décodage, soit au même pour l'étude du caractère suivant dans la chaine-format.



Organigramme général de l'éxécution de la procédure S O R T I R

1.4 Organisation de l'étude des formats primaires

On détaille dans ce paragraphe les points 9 et 10 du § 1.2 .Il constitue en fait la légende de l'organigramme de la page ci contre

-1 - Si le primaire commence par un alignement, l'effectuer(LIGNE) et retourner à 1

2 - Compter le nombre de positions occupées au niveau externe par le détail de format (APEL1) et éventuellement changer de ligne(LIGNE)

3 - Etude caractère par caractère (APEL2) du détail de format. A la rencontre de

- i) une marque d'alignement ou une virgule: aller à 5
- -ii) une insertion: la composer (ESPACE,ARABES) et retourner à 3
- -iii)N: aller à 4
- -iv) le début d'un format booléen: aller à 6
- - v) le début d'un format numérique: aller à 7
- - v) la fin de la chaine : aller à 10

4 - Appel de la donnée courante(ELSV) et composition sous forme normalisée (NORMA)

5 - Etude caractère par caractère de la fin de l'élément de format(APEL 4) à la rencontre de

- i) une marque d'alignement: faire l'opération désirée (LIGNE) et retourner à 5
- ii)Une insertion : composition (ARABES, ESPACE); retourner à 5
- -iii)Une fin de chaine : aller à 10
- iv)Une virgule : aller à 10
- v) Autre chose: erreur. Se brancher à SECOUR qui la signalera et appellera NORMA pour la sortie des autres données

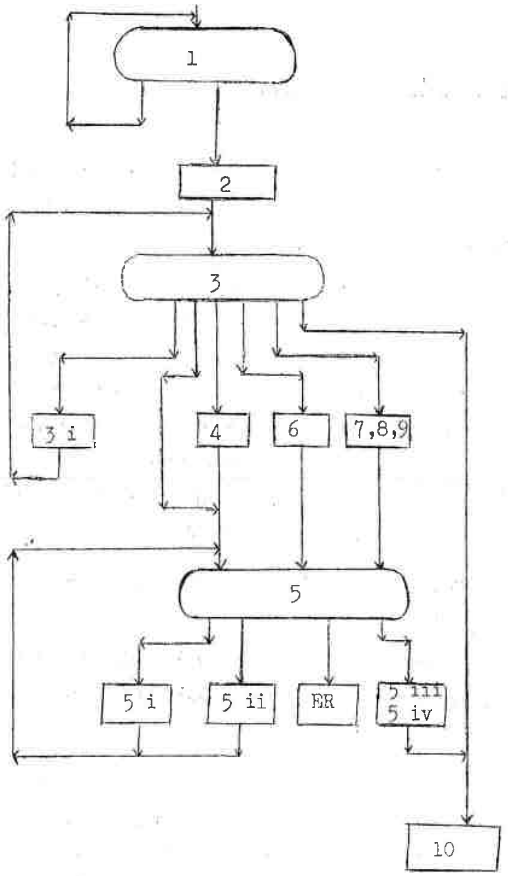
6 - Appel de la donnée booléenne(ELSV) et composition en respectant le format (LOGIQ, BOOL); aller à 5

7 - Etude syntaxique du format de nombre et décompte des valeurs L_i

8 - Appel de la donnée courante (ELSV)et initialisations

9 - Etude et composition caractère par caractère (APEL3) du format de nombre en fonction des valeurs des L_i et de celle du nombre (NOMBRE); aller à 5

10 - Cette instruction est en fait celle numéro 11 de l'organigramme de la page 85



Organigramme de l'étude d'un format primaire - Légende p.86

1.5 remarques sur les programmes de servitude

Deux ont déja été décrits au chapitre précédent: ELSV et ESTYP. Ce dernier effectue lorsqu'il est appelé par SORTIR le choix de la ligne de composition (voir ci dessous, § 2.2).

SECOUR signale les erreurs, appelle s'il le faut la donnée courante et la sort sous forme normalisée.

Le sous-programme DEMUL fera l'objet du paragraphe 4.

2 Sous-programmes d'étude des formats

2.1 Sous-programmes de décodage

Ce sont ceux permettant de parcourir une chaine en étudiant ses caractères les uns après les autres, chacun conduisant à un traitement spécial selon sa nature.

La position d'un caractère est repérée par un compteur(SE) qui progresse à chaque appel du sous-programme CTFOR. Ce sous-programme se branche à un autre sous-programme en fonction de la nature du caractère en cours d'étude. Pour cela on utilise une table, FORTAB, dont les arguments sont les caractères pouvant composer une chaine format , et ses fonctions les adresses de ces sous-programmes. En fait la table possède 6 " colonnes" d'adresses: 4 pour les appels de décodage APEL1 à APEL4, une pour SORTAB et une pour FORMAT.

Nous allons voir la gestion de ces sous-programmes en traitant un exemple:

Soit le primaire /'X= 'DDBB/D (qui est volontairement erroné). La partie de la table FORTAB qui nous intéresse est approximativement la suivante:

B	ZONE1	ESPACE	ESPACE	ESPACE
'	ARA1	ARABES	ARABES	ARABES
/	VIRG1	APEL4	APEL4	LIGNE
D	ZONE1	NOMBRE	ZER3	ER

La première barre / est traitée lors du point 1 du §1.4 et conduit à l'appel du programme LIGNE, qui se branche au premier décodage APEL1. Le compteur SE indique alors l'adresse du guillemet ouvert. On utilise la table ci dessus par l'intermédiaire du sous-programme CTFOR: le sous-programme se trouvant dans la colonne 1 en face de l'argument ', en l'occurence ARA1, est appelé. Il fait progresser SE régulièrement. Si le caractère à cette adresse n'est pas le guillemet fermé de l'insertion, un compteur, LONG1, croit de un. Si c'en est un, ARA1 retourne à CTFOR Dans cet exemple, LONG1 a pris la valeur 4.

CTFOR fait progresser SE, trouve le premier D et se branche en ZONE1. Là on ajoute un à LONG1 et on revient à CTFOR, qui fera la même chose avec le D et le B suivants. LONG1 prend ainsi la valeur 8.

La barre / est découverte et on se branche en VIRG1: on a fini APEL1,i.e.de parcourir le détail de format: on appelle AVZS(avec LONG1 comme argument) pour savoir si les huit caractères tiendront dans la ligne (si tel n'avait pas été le cas, la ligne aurait été imprimée et la composition se continuerait au début d'une nouvelle ligne). On affecte à SE l'adresse du guillemet ouvert et on va en APEL2.

Ce seconde décodage appelle à son tour CTFOR qui se branchera maintenant aux adresses de la seconde colonne: la première trouvée est ARABES. Ce sous-programme compose les insertions: il fait progresser SE régulièrement, envoyant chaque caractère de l'insertion dans la ligne, sauf si c'est le guillemet fermé auquel cas on revient à CTFOR. Celui ci trouve alors le premier D et va en NOMBRE où sera traité le format numérique.

Nous reviendrons plus bas sur ce sous programme. Signalons qu'il se termine par l'emploi du décodage APEL3 qui utilise CTFOR et la troisième colonne de FORTAB: à chaque D on va en ZER3 qui envoie un zéro ou un chiffre significatif dans la ligne de composition. Le B détecté, on va en ESPACE qui met simplement un blanc dans la ligne, et se rebranche en CTFOR. Le second B produit la même opération. CTFOR trouve alors la barre indiquant que l'on sort du détail de format pour entrer dans un élément de format: on va donc au quatrième décodage.

Celui ci utilise à son tour CTFOR mais la quatrième colonne de FORTAB. LIGNE est appelé par la barre; il éxécute l'opération d'alignement demandée et retourne en CTFOR qui trouve le D: on se branche alors à ER qui signalera une erreur (en effet la syntaxe interdit des marques d'alignement à l'intérieur d'un format de nombre). Désormais toutes les autres données internes qui restent éventuellement à sortir sont appelées par ELSV et traitées par NORMA qui les compose sous forme normalisée. Signalons que si ce dernier D était remplacé par une virgule on retournerait à APEL1 pour étudier le format suivant et si c'était le guillemet fermé de la fin de la chaine format, on irait à FINS où divers compteurs seraient forcés avant le retour au programme objet en RETOU.

2.2 Gestion de la ligne de composition

Il existe en fait deux lignes de composition : l'une pour les sorties sur cartes, l'autre pour celles sur machine à écrire.

Lors du chargement du programme, ces lignes sont mises sous la forme de lignes vides. Un système de compteurs permet, par des initialisations lors de chaque appel de SORTIR, de faire les opérations d'alignement et de composition comme si il n'y avait qu'une seule ligne.

Lorsque le programme arrive au dernier FIN du programme Algol, ces lignes seraient imprimées si elles n'étaient pas vides.

2.2.1 Opérations d'alignement

Le sous-programme LIGNE est appelé chaque fois que l'on doit changer de ligne, soit qu'une marque d'alignement ait été spécifiée dans le format, soit que le nombre de positions externes d'un détail de format ne puisse tenir dans la ligne actuelle.

Si la sortie se fait par l'unité 1622, une carte contenant la ligne est perforée, puis la ligne est remise à blanc de façon qu'un appel immédiat de LIGNE sorte une carte vierge.

Si la sortie se fait sur machine à écrire, deux cas peuvent se présenter :

- la ligne n'est pas vide : on tape les caractères qui la composent, on fait un retour charriot et on remet cette ligne à blanc.
- la ligne est vide : on ne fait qu'un retour charriot.

Le sous-programme SAUTY est chargé du traitement de l'opération de tabulation (code format S). Si la sortie se fait sur cartes, on ne tient pas compte de ce format, sinon, on fait les mêmes opération que pour LIGNE, mais en remplaçant "retour charriot" par "tabulation".

2.2.2 Le sous-programme AVZS

Il est appelé chaque fois que l'on veut savoir si une séquence de N caractères pourra tenir dans la ligne en composition. Si cela était impossible, le sous-programme LIGNE serait appelé.

2.3 Etude des formats de nombre

Cette étude nécessite deux phases :

- vérifications syntaxiques
- composition caractère par caractère

2.3.1 Etude de la syntaxe d'un format de nombre

Les nombres doivent apparaître au niveau externe comme des nombres ALGOL (insertions mises à part). Il n'est donc pas surprenant que la grammaire des formats de nombre soit du même type que celle des nombres ALGOL, et que l'on puisse la décrire par une grammaire de Kleene semblable à celle du chapitre 1 :

En effet, on peut écrire par exemple :

⟨format de nombre sans signe⟩ ::= Z ⟨format de nombre sans signe⟩|
D ⟨format de partie significative⟩|
. ⟨format de partie décimale⟩|
E ⟨format d'exposant avec signe⟩|
E ⟨format d'exposant sans signe⟩|
Z | D

⟨format de partie significative⟩::= D ⟨format de partie significative⟩|
. ⟨format de partie décimale⟩|
E ⟨format d' exposant avec signe⟩|
E ⟨format d' exposant sans signe⟩|
D

Ces règles étant calquées sur celles de la grammaire des nombres en faisant précéder chaque terme du vocabulaire par le mot "format". En fait, quelques compléments sont nécessaires :

- possibilité de signe avant ou après le nombre
- absence de partie décimale pure
- possibilité d'insertions (qui, en fait, sont ignorées, comme l'était le symbole ␣ pour les nombres).

Pour analyser ce langage, dont on ne juge pas utile de donner la grammaire complète, on utilise encore un automate fini du même type que ceux utilisés pour la compilation des nombres et pour l'entrée des données numériques.

Toutefois, les opérations liées à la sémantique que l'on effectue au passage d'un état à l'autre ne sont plus les mêmes. Ici, elles sont du type : faire progresser L_i de un, la valeur i étant déterminée par l'état où l'on se trouve.

Tout passage d'un état à un autre qui ne soit prévu par la grammaire révèle l'existence d'une erreur syntaxique dans l'écriture du format de nombre ; un libellé d'erreur est imprimé et le nombre sera composé par un format normalisé.

2.3.2 Composition proprement dite

A la fin de la phase précédente, on sait sous quelle forme (décimale, ou avec exposant), sera sorti le nombre, et si le nombre sera avec ou sans signe.

Nous allons expliquer la suite des opérations sur un exemple où pratiquement tous les cas seront vus de façon à ne pas trop nuire à la généralité :

soit l'élément de format :

'X = '-ZZZZDBDDDB.DDBDDDBE-DD/

qui a déjà été étudié par APEL1 (les caractères tiendront donc tous dans la ligne) puis par APEL2 qui aura traité l'insertion avant d'appeler NOMBRE. La première phase aura vérifié que la syntaxe du nombre est correcte, aura détecté la présence du E et du signe -, et enfin aura affecté aux L_i les valeurs suivantes :

$L_1=4$, $L_2=4$, $L_3=5$, $L_4=0$, $L_5=2$.

On appelle alors, par le sous-programme ELSV, l'élément de la table qui doit sortir sous ce format. Admettons que ce soit le réel :

$$A = -0.1234567899_{10}7$$

C'est bien un nombre, et son signe doit être sorti. On peut donc continuer les opérations (si ce n'était pas un nombre ou si c'était un nombre négatif, alors qu'on ne demande pas de signe, une sortie normalisée serait imposée).

Ce nombre doit sortir sous la forme d'un nombre avec exposant. On a vu au § 2.3 du chapitre 6 qu'il faut faire la composition en deux phases, la première pour la partie décimale, la seconde pour celle exposant.

On sort donc d'abord, sous forme de nombre décimal, la mantisse affectée de la caractéristique $CC'=10-L_3=5$.

Il nous faut connaître, en fonction des règles données au § 2.3 du chapitre 6, le nombre NBB de positions Z qui seront blanches, et celui NBZ de cases D qui seront remplies par des zéros. On a :

$$NBB = \begin{cases} L_1 + L_2 - CC & \text{si } CC \geqslant L_2 \\ L_1 & \text{si } CC < L_2 \end{cases}$$

$$BNZ = \begin{cases} 0 & \text{si } CC \geqslant L_2 \\ L_2 - CC & \text{si } CC < L_2 \end{cases}$$

Ici, où $CC > L_2$, on trouve NBB = 3 ; NBZ = 0 .

Ces calculs faits, on appelle le décodage APEL3 dont nous avons déjà donné le principe : à chaque caractère de l'élément de format, on se branche à un sous-programme qui fera un traitement aproprié et retournera, en général, au décodage (en CTFOR) pour étudier les caractères suivants. Ici, on effectue les opérations suivantes :

au signe - test NBB > 0? Oui, envoyer un blanc dans la ligne et baisser NBB de 1 (donc NBB = 2)

Z même chose, NBB valant 1

Z même chose, NBB valant 0

Z NBB=0. On envoie le signe dans la ligne. NBB:=NBB-1=-1

Z NBB < 0. Cette fois, on envoie le premier chiffre significatif (c'est à dire le 1) dans la ligne.

D NBZ < 0? Oui, envoi du digit suivant, c'est à dire 2

B envoi d'un blanc

D même chose que l'avant dernière fois, et de même pour les DDB qui le suivent.

. envoi d'un point dans la ligne

Le reste des B et D se traite de la même façon.

On a donc composé la ligne :

		-	1	2		3	4	5	.	6	7	8	9	9	

APEL3 trouve alors le E qui est envoyé dans la ligne. Mais cette fois, on ne retourne pas tout de suite à CTFOR : on calcule l'entier qui doit servir d'exposant pour ce nombre : on lui trouve la valeur 2, et on lui associe la caractéristique 1. On affecte ensuite à L_1 et à L_2 les valeurs respectives de L_4 & L_5 (0 et 2), et on calcule NBB(0) et NBZ(1).

On retourne à APEL3 qui recommencera le même genre d'opérations que précédemment :

Au signe + NBB = 0 envoi du signe plus
D NBZ > 0 envoi d'un zéro. Baisser NBZ de 1
D NBZ = 0 envoi du digit 2
/ on se branche en APEL4 : le format de nombre est fini.
On a composé la ligne :

X	=					-	1	2	3	4	5	.	6	7	8	9	9	E	+	0	2

qui ne sera imprimée que lorsque LIGNE sera appelé (ici, lors du décodage APEL4).

3. SORTIE D'UN TABLEAU : Exécution de la procédure SORTAB

Le sous-programme d'éxécution suit rigoureusement le programme que nous avons donné au chapitre 6. Nous n'indiquons donc ici que la façon dont on utilise les sous-programmes d'éxécution de la procédure SORTIR.

1 - Initialisations ;
2 - Etude du numéro de l'organe et choix de la ligne de composition (ESTYP)
3 - Appel du second élément de la table (ELSV) qui est obligatoirement une chaîne format
4 - Si cette chaîne ne contient pas de multiplicateurs, aller à 6
5 - Remplacer cette chaîne par la chaîne équivalente (DEMUL)
6 - Séparer les deux primaires, compter les positions Li et Le, et vérifier la syntaxe (CTFOR et 6ème colonne de SORTAB)
7 - Appel du vecteur de renseignement (ELSV). Etude de
i) la dimension N
ii) le nombre Q de composantes de la partie droite
iii) des bornes a_i, b_i de chaque indice
8 - Affecter à P sa première valeur
9 - Calcul des I_j - sortie de la série des indices (NOMBRE)
10 - Sortie des P composantes du tableau (NOMBRE ou BOOL)
11 - Tests sur les bornes et les valeurs de P - aller à 9 (ou à 12 si le tableau est sorti entièrement)
12 - Retour au programme objet à l'adresse RETOU.

Les sous-programmes NOMBRE, BOOL peuvent être utilisés par un jeu de bascules tel qu'ils n'appellent pas le sous-programme ELSV (mais prennent les valeurs des bornes ou des composantes du tableau) et reviennent au sous-programme des tableaux.

En cas d'erreur, au cours de la sortie d'un indice ou d'une composante, toutes les composantes qui ne sont pas encore sorties sont composées sous forme normalisée.

4. TRAITEMENT DES CHAINES AVEC MULTIPLICATEURS

Le programme DEMUL a pour fonction de remplacer une chaîne avec multiplicateurs par la chaîne équivalente sans multiplicateurs qui sera placée dans la zone PIF. Ce programme peut être appelé par le sous-programme d'éxécution de SORTIR (voir § 1.2) ou de SORTAB (§ 3) ou par celui de la fonction FORMAT. Dans ce dernier cas, certains multiplicateurs sont dynamiques, et il faut affecter une valeur numérique. On verra au § 4.3 comment cette affectation a lieu. Pour le moment, on ne s'occupe donc que de multiplicateurs numériques.

4.1 Principe de la démultiplication

On utilise deux piles placées l'une en face de l'autre dans la zone PIF. La première gérée par le compteur PO recevra les multiplicateurs et les parenthèses ouvertes. Dans l'autre, gérée par PV, n'entreront que les caractères devant constituer la chaîne définitive. Les opérations se font de la façon suivante (voir organigramme 3).

1 - Etude caractère par caractère (par le sous-programme CTFOR et la cinquième colonne de la table FORTAB) de la chaîne initiale. A chaque caractère on se branche à un sous-programme spécial. Si ce caractère est :
2 - Un multiplicateur, il entre dans PO
3 - Un guillemet ouvert, il entre dans PIF ainsi que tous les caractères qui le suivent jusqu'au guillemet fermé inclus.

4 - Une parenthèse ouverte, elle entre dans PO et on repère l'adresse de PIF où sera envoyé le prochain caractère

5 - Une parenthèse fermée et une virgule dans PIF. Puis :

i) Si les deux derniers caractères entrés dans PO sont des parenthèses ouvertes, on supprime la dernière

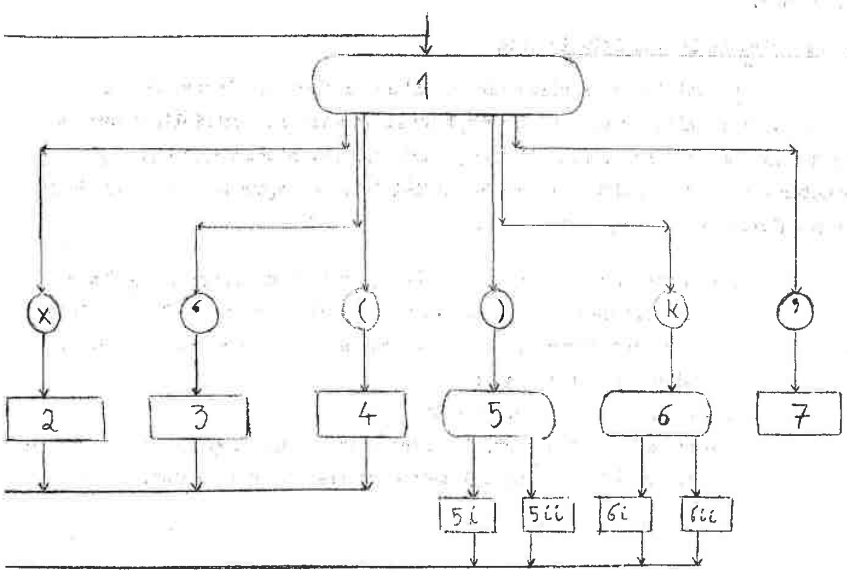
ii) Si ce sont une parenthèse ouverte et un multiplicateur, on fait la démultiplication dans la zone PIF de toute la zone comprise entre le caractère dont on a repéré l'adresse en 4 et le dernier entré, et on supprime ces deux caractères dans PO

6 - Un autre caractère par exemple Z

i) Si le dernier caractère entré dans PO est une parenthèse ouverte, ce caractère entre dans PIF

ii) Sinon (c'est un multiplicateur), on fait la démultiplication de ce caractère et on supprime le multiplicateur dans PO.

7 - Le guillemet fermé de la chaîne, on sort de DEMUL



4.2 Détails pratiques

- Les erreurs de syntaxe liées aux multiplicateurs ont été détectés dès la compilation
- L'entrée d'une parenthèse ouverte dans PO revient à entrer directement l'adresse de la première position libre de PIF.
- Un jeu de bascule permet de traiter le cas où les multiplicateurs ont un ou deux chiffres et d'éliminer (avec impression d'une erreur) ceux où ils en ont trois ou plus.
- La démultiplication se fait directement dans la zone PIF. Elle s'effectue par des transferts d'enregistrement et des calculs d'adresse. Le cas où le multiplicateur est nul est traité séparément.
- La détection d'une parenthèse fermée n'envoie pas systématiquement une virgule dans la zone PIF, notamment lorsqu'il y en a déjà une.
- Les tests sur la présence dans PO d'une parenthèse ouverte ou d'un multiplicateur sont basés sur le nombre de positions occupées par le dernier entré (2 pour un multiplicateur, 5 pour une parenthèse).

4.3 Cas des multiplicateurs dynamiques

La compilation n'admet des X dans une chaîne format que dans le cas où la chaîne est le premier paramètre de la procédure FORMAT.

A l'exécution, le cas 2 du paragraphe précédent se divise en deux branches : si le multiplicateur est numérique, il entre dans PO. Si c'est un X, on appelle le sous-programme ELSV qui affecte à cet X sa valeur numérique dont on connaît l'adresse grâce à la table d'information fournie à l'appel de la procédure FORMAT.

4.4 Remarque

Les chaînes formats constituent un langage context free. C'est ce qui nous a conduit à utiliser deux piles pour le traitement des multiplicateurs.

Chapitre 10

ERREURS SIGNALEES LORS DE LA COMPILATION ET DE L'EXECUTION DES PROCEDURES D'ENTREE-SORTIE

Nos programmes de compilation et d'éxécution ont été conçus de façon que l'on puisse signaler le maximum d'erreurs et que les programmes d'erreurs soient accessibles sans perte de temps ou de place et permettent de donner le maximum d'information.

1. ERREURS DETECTEES A LA COMPILATION

Elles sont signalées par notre programme IMPER qui, dans l'état actuel du compilateur, interdit de poursuivre la compilation (cf Annexe de l). Le libellé comprend les deux lettres ER suivies d'un numéro d'ordre: Dans les exemples que nous donnerons nous supposerons que E est un entier, R un réel, B un booléen, T un tableau à deux dimensions et ETI une étiquette.

n°	Signification	Exemples
79	Nombre incorrect de virgules	ENTRER(2,,E)
80	Parametre impossible	ENTRER(2,ETI)
81	Le numéro d'organe n'est pas une expression arithmétique	SORTIR(B,E)
82	Un paramètre de la liste d'entrée n'est pas valable	ENTRER(2,COS(R))
83	Un paramètre de la liste de sortie n'est pas valable	SORTIR(2,T)
84	Une chaine format apparait ailleurs que comme second paramètre	SORTIR(2,'A+',DD', E)
85	Le second paramètre de SORTAB n'est pas une chaine	
86	Le troisieme paramètre de SORTAB n'est pas un identificateur de tableau	SORTAB(2,'2N', T[I,J])
87	Il y a plus de trois paramètre dans un appel de SORTAB	

n°	Signification	Exemples
88	Il manque au moins un paramètre pour FORMAT ou SORTAB	FORMAT('DD')
89	Le premier paramètre de FORMAT n'est pas une chaine	FORMAT(2,'XD',E)
91	Le second paramètre de FORMAT n'est pas une expression arithmétique	
92	Un caractère X est utilisé dans une chaine en dehors de FORMAT	SORTIR(2,'XD',E)
93	Le nombre d'expressions arithmétiques dans la liste de format n'est pas égal au nombre de X dans la chaine correspondante	FORMAT('XBXD',E)
94	Une chaine n'est précédée ni d'une virgule ni d'une parenthèse	
95	Une chaine n'est suivie ni d'une virgule ni d'une parenthèse	
96	Erreur de syntaxe dans une chaine: un multiplicateur est suivi d'un caractère non valable	'XB X'E''
97	Caractère invalide dans une chaine	'4DC3B'
98	Chaine trop longue vu l'état des piles O et V	
99	Chaine de plus de 200 caractères	

Remarques

- Pratiquement toutes ces erreurs sont détectées et différenciées grâce à la consultation de table signalée au chapitre 7.

- L'erreur 98 n'est pas une erreur du programmeur, mais est équivalente à une saturation de la machine: les chaines sont, rappelons le, stockées momentanément entre les piles O et V. Si celles ci sont très pleines, il peut ne plus rester assez de place entre leurs sommets pour y stocker une chaine.

- Il ne nous est pas possible actuellement de détecter les erreurs en rapport avec la correspondance entre la chaine format et la liste de sortie.

C'est ainsi que l'appel

```
entier E;
SORTIR(2,'4F',E);
```

qui est erroné puisque l'on demande la sortie de la valeur numérique E avec un format booléen, sera compilé normalement. Bien entendu, cette erreur sera détectée et signalée à l'éxécution.

2 ERREURS DETECTEES AU COURS DE L'EXECUTION DE "ENTRER"

Ces erreurs sont imprimées par notre programme BUTOIR sous la forme du libellé ER E suivi d'un numéro:

n°	Signification	Exemples
2	Faute dans l'écriture d'une donnée booléenne	VRAIbCARTEb...
3	Un entier est plus grand (en valeur absolue) à 10^{12}, ou un réel à 10^{99}	.Z14 - E 110
4	Faute de syntaxe dans une donnée numérique	-12..34 .80Ebbb17
5	Entrée sur machine à écrire d'une ligne de plus de 80 caractères	
6	Entrée sur machine à écrire d'une ligne de plus de 160 caractères	

2.1 Les erreurs E 5 et E 6

Le programmeur n'a pas suivi la restriction que nous avons imposée sur le nombre de caractères pouvant entrer par ligne par machine à écrire. Deux cas peuvent se présenter:

-i) Il y a eu entre 80 et 160 caractères: on ignore ceux qui suivent le quatre vingtième et l'éxécution se poursuit normalement (mais il risque de se produire ensuite une erreur E 2 à E 4).

-ii)Il y a eu plus de 160 caractères entrés: les sous-programme d'éxécution peuvent avoir été détruits. Toute exécution est désormais aléatoire: on préfère l'arrêter: BUTOIR tape le libellé suivant: EXECUTION IMPOSSIBLE, et le programmeur ne peut passer outre. Il est alors nécessaire de tout recharger. C'est le seul cas où une erreur d'entrée sortie ne puisse être corrigée.

.2Correction des erreurs E2 à E4

Après l' impression du libellé, la machine à écrire tape la dernière ligne entrée, puis met une astérisque sous la dernier caractère étudié qui est celui erroné dans le cas des erreurs E2 ou E4 , ou le premier qui suit la donnée numérique dans le cas de l'erreur E3.

Trois possibilités sont laissées au choix du programmeur:

-i) Correction de la donnée erronée: La machine à écrire est sélectée , et on peut modifier une partie de la ligne après avoir indiqué à partir de quelle position se fait la correction.

-ii) Recommencer l'entrée de toute la file, c'est à dire le dernier appel. Il suffit de taper une " marque d'enregistrement".

-iii) recommencer toute l'éxécution du programme.

3. ERREURS DETECTEES A L'EXECUTION DES PROCEDURES DE SORTIE

Toutes ces erreurs sont également signalées par le programme BUTOIR et sont corrigées par le programme de façon que l'éxécution puisse se continuer. Le libellé est ER S suivi d'un numéro:

n°	Signification	Exemples
1	Un détail de format demande plus de 80 caractères au niveau externe	SORTIR(2,'85B')
2	Ie nombre de formats numériques ou booléens est supérieur au nombre d'éléments de la liste de sortie	SORTIR(2,'DD,4F',E)
3	Ie nombre de formats numériques ou booléens est inférieur au nombre d'éléments de la liste de sortie	SORTIR(2,'DD',E,B)
4	A un format booléen correspond une variable booléenne	SORTIR(2,'L',E)
5	A un format numérique correspond une variable booléenne	SORTIR(2,'4D',B)
6	Nombre trop grand pour sortir en respectant le format numérique	SORTIR(2,'DD',123)
7	Erreur de syntaxe dans un élément de format	SORTIR(2,'BB/N',E)
8	Sortie de titre impossible: la zone PIF a été utilisée depuis l'entrée du titre à reproduire.	

n°	Signification	Exemples
9	Erreur de syntaxe dans la partie décimale d'un format numérique	SORTIR(2,' ZZ.Z',E)
10	Erreur de syntaxe dans la partie exposant d'un format numérique	SORTIR(2,'4D.DEDD',E)
11	Erreur dans un format booléen	SORTIR(2,'FF',B)
12	Multiplicateur écrit avec trois chiffres	SORTIR(2,'010D',E)
13	Multiplicateur X associé à une variable numérique	
14	Multiplicateur X associé à une nombre plus grand que 99	
15	Saturation de la machine:il y a trop de multiplicateurs dans la chaine format	
17	Il n'y a pas deux primaires dans le format pour SORTAB	SORTAB(2,'BB,DD,BDD',T)
18	Tous les indices d'un tableau ne peuvent tenir sur une seule ligne	SORTAB(2,'50DD,BD',T)
19	Les indices tiennent trop de place pour qu'il en reste pour une seule composante	SORTAB(2,'5B25D,3D20B',T)
20	Opération d'alignement demandée dans SORTAB	SORTAB(2,'DD,BDD/',T)

3.1 Auto-correction par non opération

Les erreurs S2 et S7 sont signalées, et on ne tient pas compte de ce qui est en trop. Exemple: SORTIR(2,'4D,4F',E) E sortira sous le format 4D et celui 4F sera ignoré.

3.2 Sortie normalisée

Toutes les autres erreurs (sauf E 17 et suivantes) conduisent à une sortie normalisée: l'élément de liste sort avec un format normalisé ainsi que tous les suivants du même appel.

Exemple: l'appel SORTIR(1,'/4DBB,FBB,3D',1234,-1.0,123) produira les lignes suivantes:

```
ER S 4
1234  -.1000000000E 01          123
```

3.3 Sortie normalisée de tableau

Dès la détection d'une erreur, les composantes qui ne sont pas encore sorties sont mises sous forme normalisée. Mais les indices ne sortent plus.

3.4 Erreurs de multiplicateurs

S 12 : sortie normalisée

S 13 : on prend la valeur absolue de X

S 14 : X est traité modulo 100

3.5 Erreurs locales et définitives

On appelle erreur locale une erreur qui peut n'apparaître qu'au moment d'un appel particulier, et qui pourra ne pas se reproduire. Le libellé d'erreur n'apparait qu'à chaque appel erroné. Exemple:

pour E:= 12,13,144,15 faire SORTIR(1,'/DD',E);donnera les lignes:

12
13
ER S 6
144
15

Lorsque au premier appel on détecte une erreur qui se répètera à chaque appel, le libellé correspondant est imprimé, la sortie se fait sous forme normalisée. Mais à chaque nouvel appel, la sortie sera normalisée mais il n'y aura plus d'impression du libellé. Exemple:

pour E:=1,2,3 faire SORTIR(1,'/4F',E) donnera:

ER S 4
1
2
3

3.6 Remarques

Plusieurs erreurs peuvent être signalées successivement lors d'un même appel. C'est notamment le cas si un multiplicateur est inférieur à -99 puisque les erreurs S 13 et S 14 ne conduisent pas à une sortie normalisée.

Chapitre 11

EXEMPLES DE S O R T I E S

Nous donnons dans ce chapitre quelques exemples de sorties non spécifiquement numériques pour montrer l'intérêt et les possibilités de sorties " étendues". Ce que nous voulons surtout montrer c'est la facilité d'écriture des algorithmes.

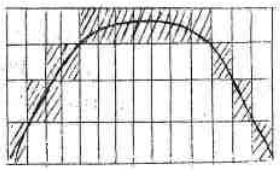
Bien qu'il nous arrivera de nous servir de la procédure FORMAT tous ces exemples auraient pu être traités sans cette fonction. En effet ces possibilités graphiques sont essentiellement dues au fait que SORTIR est une instruction de composition et non d'impression.

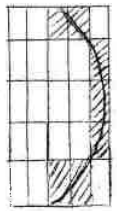
Plusieurs de ces exemples reviennent à considérer que l'on sort une grille dont on remplit les lignes case par case. la grille ainsi formée a toutefois deux inconvénients:

- on ne dispose que de 80 colonnes, et les échelles horizontales et verticale ne sont pas les mêmes.

- la grille étant petite, la " définition" d'une courbe ne pourra qu'être approchée.

C'est ce qui explique qu'une courbe ne pourra être bien rendue au voisinage des tangentes horizontales ou verticales. Exemple:





Mais, rappelons le, le principe nous intéresse d'avantage que les résultats dans ce chapitre.

1 TRIANGLE DE PASCAL

Nous utilisons un algorithme de [12] pour donner un exemple de présentation de résultats numériques basée sur l'utilisation des opérations d'alignement et de composition:

```
début entier K,N; entier tableau A[0:13];
     SORTIR(2,'' TRIANGLE DE PASCAL'//');
     pour N:=0 pas 1 jusqua 13 faire
     début  A[N]:=1;
            pour K:=N-1 pas 1 jusqua 1 faire  A[K]:=A[K-1] + A[K];
            SORTIR(2, FORMAT('//XB', 39-3xN));
            pour K:=1 pas 1 jusqua N faire SORTIR(2,'3B3Z', A[K])
     fin
fin
```

Ce qui nous a donné les résultats de la page ci-contre.

Remarques

A la place de l'instruction SORTIR(2,FORMAT('//XB',39-3xN)) on aurait pu écrire:

```
SORTIR(2,'/');
pour K:=1 pas 1 jusqua 13-N faire SORTIR(2,'3B');
```

Si on avait voulu sortir une ligne supplémentaire avec le même format numérique '3B3Z', la quatorzième donnée de cette ligne imposerait une opération d'alignement (puisque 14 x 6 > 80) c'est à dire qu'elle sortirait sur une autre ligne.

TANGLE DE PASCAL

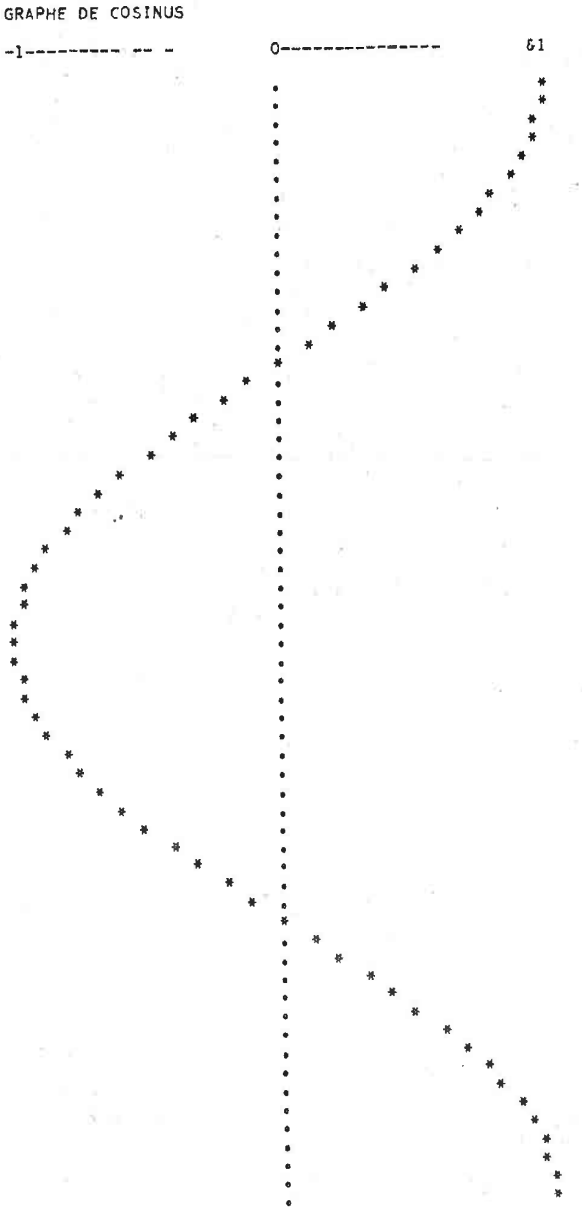
```
                                1
                              1   1
                            1   2   1
                          1   3   3   1
                        1   4   6   4   1
                      1   5  10  10   5   1
                    1   6  15  20  15   6   1
                  1   7  21  35  35  21   7   1
                1   8  28  56  70  56  28   8   1
              1   9  36  84 126 126  84  36   9   1
            1  10  45 120 210 252 210 120  45  10   1
          1  11  55 165 330 462 462 330 165  55  11   1
        1  12  66 220 495 792 924 792 495 220  66  12   1
```

2 Sortie d'une courbe

Nous voulons sortir le graphe de COS(X) sans utiliser la fonction FORMAT, et en indiquant l'axe dù X vaut æéro.

```
Début entier I,CX; réel X, DELTAX, DEUXPI;
     DEUXPI:=6.283185306; DELTAX:=DEUXPI/60;
     SORTIR(2,''GRAPHE DE COSINUS'//');
     SORTIR(2,' -1', 2Ж(' -'),'0', 2Ж(' -'), '1'/');
     pour X:=0 pas DELTAX jusqua DEUXPI faire
     début CX:=25xCOS(X);
          si CX = 0 alors SORTIR(2,'25B'*'/');sinon
          si CX > 0 alors
                  début SORTIR(2,'25B'.');
                        pour I:= 1 pas 1 jusqua CX-1 faire SORTIR(2,'B');
                        SORTIR(2,'*'/')
                   fin
           sinon   début pour I:= 1 pas 1 jusqua 25+CX faire SORTIR(2,'B');
                        SORTIR(2,'*');
                        pour I:= 1 pas 1 jusqua -(CX+1)faireSORTIR(2,'.'/')
                   fin
     fin
fin
```

Nous avons obtenu ainsi la courbe ci-contre.



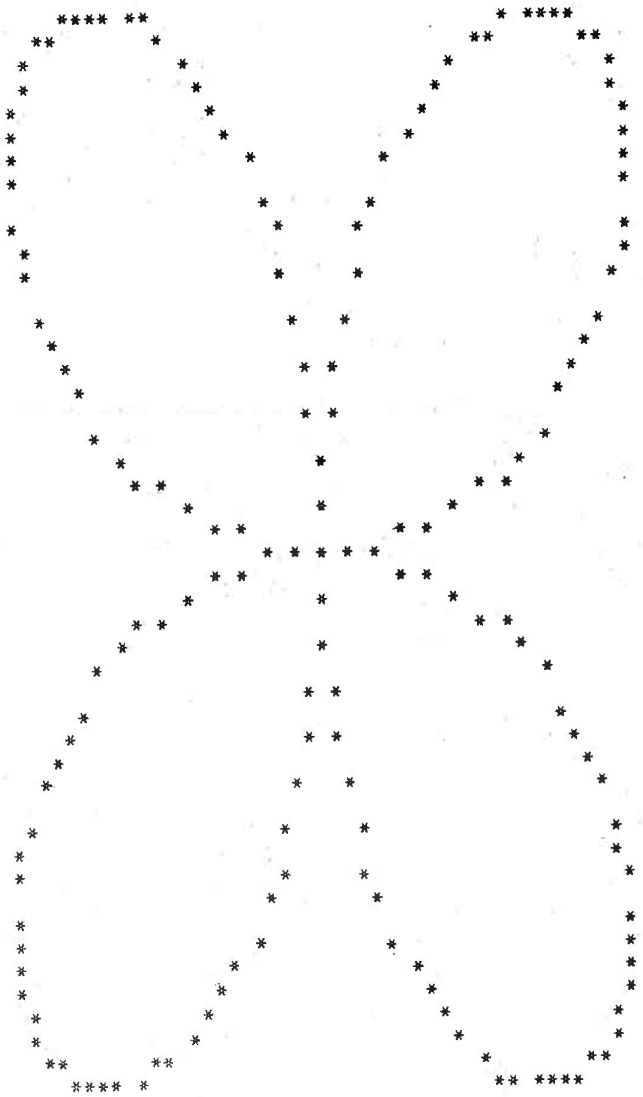
Remarques

Si on voulait utiliser FORMAT et ne pas imprimer l'axe des X, le premier bloc pourrait être remplacé par l'instruction unique:

```
SORTIR(2, FORMAT('XB'*'/',25(1+COS(X)))
```

Un tel tracé ne serait pas lisible lorsque la fonction présente des extrema très accusés, ce qui peut être le cas de " spectres". Il suffit alors, dans cette dernière instruction, de remplacer le format 'XB'*'/' par 'X('*'),'*'/'.

Si enfin on désire avoir un axe des X sur une horizontale, on peut facilement le faire en utilisant un tableau.



3 Sortie d'une courbe définie paramétriquement

Le principe est de calculer les abscisses et ordonnées de quelques points de la courbe. Pour cela il suffit d'utiliser une matrice booléenne.

Nous donnons ici un exemple de courbe définie par l'équation polaire $\rho = 30 \sin 2\theta$.

```
début entier I,J;réel S2X,DEUXPI, TETA, DELTA;
    booléen tableau A[-25:25, -25:25];
    DEUXPI:= 6.283185306; DELTA:= DEUXPI/180;
    pour I:= -25 pas 1 jusqua 25 faire
    pour J:= -25 pas 1 jusqua 25 faire A[I,J]:= vrai;
    pour TETA:= 0 pas DELTA jusqua DEUXPI faire
    début S2X:= 30 x SIN(2xTETA);
          A[S2X x SIN(TETA) , S2X x COS(TETA)] := faux
    fin
    pour I:= -25 pas 1 jusqua 25 faire
    début SORTIR(2,'/');
          pour J:= -25 pas 1 jusqua 25 faire
          si A[I,J] alors SORTIR(2,'B') sinon SORTIR(2,"*")
    fin
fin
```

Ce programme nous a donné le résultat de la page à gauche. On remarquera une distorsion verticale(due à la non égalité des échelles sur la tabulatrice), que le pas de TETA n'a pas été très bien choisi ce qui donne une définition assez mauvaise de la courbe.

Un tel programme peut être utilisé par exemple pour le tracé de d'équipotentielles dans tous les problèmes de champs.

-

ATIONS CARTOGRAPHIQUES

ralités

Le dernier exemple que nous venons de donner ne faisait somme toute rtir un tableau sous la forme d'une grille dont chaque case était occupée ar un blanc, soit par un caractère précis.

Or le matériel que nous utilisons possède environ une quarantaine de ères différents mettant ainsi à notre disposition tout un jeu de valeurs l'information que l'on donne à leur combinaison. Si on appelle carte la ainsi sortie, ce dernier exemple constitue une " carte inventaire",ce us a conduit à utiliser un tableau booléen ("il existe" et "il n'existe tant remplacés par vrai et faux). Maintenant, si on utilise un tableau on pourra faire correspondre à chaque caractère externe (ou à une aison de plusieurs d'entre eux) une valeur numérique. Ainsi par exemple pourra correspondre un blanc, à 1 un point, à 2 la barre /, à 3 la lettre O les deux lettres WW etd.... Ces relations constituent la légende des s messages" qui au niveau externe sera du type: + pour un gisement de e porphyroïde, O pour la position d'un atome d'oxygène, ou bien / a partie en acier trempé de quelque machine. Bien sûr tout cela n'est pas u, mais ce qui nous parait important c'est qu'on puisse écrire très facile- l'algorithme de sortie. Il suffit en effet d'une instruction du type:

```
si A= 0 alors SORTIR(i,'B') sinon
si A= 1 alors SORTIR(i,".")sinon
si A= 2 alors SORTIR(i,"V")sinon
  .... .
SORTIR(i,"W");
```

deux instructions POUR imbriquées(et éventuellement une instruction gnement).

Dans un tel type de légende, l'emploi de chaque caractère est purement ntionnel et il n'existe aucune relation de grandeur entre eux. Or on peut emment avoir besoin d'une correspondance (par exemple pour les cartes en de répartition démographique on emploie des caractères plus ou moins es en fonction de la densité de population). Cette fois ce que l'on peut ser ce n'est plus une différence de forme entre chaque caractère, mais

la différence entre leur surface imprimée(un W étant par exemple plus sombre qu'un O). Cette méthode non plus n'est pas nouvelle, elle est notamment utilisée en médecine par [15] qui a publié divers codes dont nous nous sommes inspirés. Mais ce qui nous intéresse ici, c'est qu'on puisse encore utiliser très facilement la procédure SORTIR, par exemple de la façon suivante:

```
si X≤0 alors SORTIR(i,'B') sinon
si X<10^2 alors SORTIR(i,"V") sinon
si X<10^5 alors SORTIR(i,"W") sinon
  etc....
```

2.Exemple de construction d'une carte radiogéochogique

Le problème est le suivant: un secteur géologique de plusieurs kilomètres carrés fait l'objet de prélèvements pétrographiques selon une maille assez serrée. Chaque échantillon est analysé en laboratoire. On étudie par exemple la teneur en MgO, CaO, Fe_2O_3,etc... puis on calcule divers coefficients. On construit alors pour chaque coefficient une carte où l'on porte les valeurs de chaque échantillon, et on trace les zones comprises entre certaines teneurs. Ces cartes sont ensuite étudiées en les comparant les unes aux autres ou à la carte géologique du secteur. Pour peu que l'on ait plus d'un millier d'échantillons et ne serait-ce qu'une dizaine de coefficients, on arrive très vite à un travail fastidieux.

Nous allons montrer sur un exemple concrêt comment ce problème peut se traiter par un programme Algol:Au secteur géologique est associéeune matrice dont les indices seront les coordonnées géographiques. Chaque composante de la matrice est affectee de la valeur zéro. On lit ensuite des cartes portant les coordonnées d'un échantillon et les résultats des diverses analyses. On affecte à la variable indicée (dont les indices sont les coordonnées géographiques) la valeur du coefficient désiré (ici ce sera $C=U_T/U_F$). La lecture achevée, il suffit de sortir ce tableau composante par composante en suivant un code.

Nous passons sous silence divers détails techniques liés aux valeurs des coordonnées, à l'échelle de la carte et aux choix des caractères utilisés pour la sortie. Nous connaissons le nombre de données. Sinon, on pourrait utiliser la méthode donnée en exemple au chapitre 5(§4).

```
ébut commentaire carte géologique du Limousin;
éel(X,Y; tableau A[1:25, 1:25]; entier I,J,K,L,UT,UF;

ommentaire Remise à zéro du tableau;
our I:= 1 pas 1 jusqua 25 faire
our J:= 1 pas 1 jusqua 25 faire A[I,J]:= 0;

ommentaire construction de la carte;
our I:= 1 pas 1 jusqua 840 faire
ébut ENTRER(2,X,Y,UT,UF);
     K:= (X-509000)/1000 ; L:= (Y-106500)/500 ; C:= UT/UF;
     si A[K,L]<C alors A[K,L]:=C;
in
ommentaire sortie de la carte;
our I:= 1 pas 1 jusqua 25 faire
ébut SORTIR(2,'/');
    pour J:=1 pas 1 jusqua 25 faire
    début K:= A[I,J];
          si K = 0 alors SORTIR(2,'B') sinon
          si K<500 alors SORTIR(2,"/") sinon
          si K<900 alors SORTIR(2,"V") sinon
          si K<1200alors SORTIR(2,"W") sinon
          SORTIR(2,"=");
    fin
  fin
fin
```

Remarques

On ne s'occupe pas dans cet essais du problème du choix des bornes des zones d'isoteneur (ici 500,900,...1500) qui relève de la statistique, ni de celui du traitement des variables nulles qui est un problème d'interpolation dans le plan .

- A N N E X E -

```
*
******    COMPILATION DES NOMBRES
*
M       DS   ,C1&9
CC      DS   ,C1&11
K       DS   ,MEMSE
DIXFX DS   ,745
CENTFXDS   ,757
ZEROFXDS   ,661
ZFL     DS   ,769
DEMIFLDS   ,781
DIXMFLDS   ,793
CENMFLDS   ,805
QUARFLDS   ,817
UNFL    DS   ,829
DEUXFLDS   ,841
CARA    DS   6
ZERO    DC   25,0,
        DC   1,0,
UN      DC   13,1000000000000,
        DC   13,5000000000000,
AR      DS   2
QUART   DC   10,2500000000,
DEUX    DC   10,2000000000,
DEBU1   SF   F2
ADRSUNDC   5,673,*
        TFM  DEBU2-6,KNUL
        TF   M,ZERO
        TFM  CC&1
        DC   3,0,*
        TR   F6,CHFL
        TF   CARA,ZERO-19
        TFM  TCH&6,M-13
        TFM  TDFX&6,CARA-3
        CF   SYMBOL
        SM   MEMSV,6,10
        B
DEBU2   TFM  *-6,PTD
        B    DEBU1&24
        DORG*-4
F2      DS   ,DEBU2-1
F9      DS   ,DEBU2-2
F6      DS   1
F7      DS   1
F5      DS   1
        DORG*&2
DEBU3   TFM  DEBU2-6,EXP
        TF   M,UN
        TFM  CC&1
        DC   3,010,*
        B    DEBU1&60
        DORG*-4
CHFL    DC   1,-0,
        DC   2,0,
        DC   1,0,
CHF     BTM  DECAL,70,10
        BL   PT
        BNF  PTD&24,F7
```

```
        BNE KNUL&12,F5
        AM  CC,1,10
        BNE CHF,F6
        CM  TCH&6,M-1
        BE  TCH&20
        AM  *&18,1,10
TCH     TD  ,K,11
        B   CHF
        DORG*-4
        CF  F6
        CF  F2
        B   TCH-12
        DORG*-4
KNUL    CM  K,70,610
        BE  CHF
        SF  F5
        B   CHF&48
        DORG*-4
PT      CM  K,3,610
        BNE LTR
        BTM DECAL,69,10
        BE  EXP
        BL  FINEX&84
        BNE SER,F7
PTD     CF  F7
        CF  F2
        BNE *&20,F5
        B   CHF&60
        DORG*-4
        CM  K,70,610
        BE  *&32
        SF  F5
        B   TCH-12
        DORG*-4
        SM  CC,1,10
        B   CHF
        DORG*-4
EXP     TFM F2,0,9
        TFM F5,0,9
        BTM DECAL,10,10
        BE  CHEX
        CM  K,20,610
        BNE *&36
        SF  F9
CHEX    BTM DECAL,70,10
        CM  K,70,610
        BL  FINEX
        BNE TDEX&20,F5
        CM  TDEX&6,CARA
        BE  SER&12
        AM  *&18,1,10
TDEX    TD  CARA-3,K,11
        B   CHEX
        DORG*-4
        SF  F7
        BE  CHEX
        SF  F5
        B   TDEX-12
        DORG*-4
DX      DS  ,TDEX&6
FINEX   BNE SER&24,F7
        CM  K,40,610
        BNL SER
        CM  K,3,610
        BNE LTR&24
        BTM DECAL,69,10
        BNL SER
        TFM NBCTE&30,SYMBOL&84
        B   LTR&36
        DORG*-4
LTR     CM  K,39,610
        BH  SER
        TFM NBCTE&30,EXIST?
        BNE FLOT,F2
        TDM MEMSV,1,6
        SM  MEMSV,1,10
        CM  M,0,10
        BE  CTEEL&120
        TFM *&30,M-24
        A   *&18,CC
        SF  M-24&CC
        AM  *-6,11,10
        CM  *-18,5,610
        BNH CTEEL&72
        CM  *-42,10,610
        BE  CTEEL&132
        CM  *-66,100,60
        BE  CTEEL&144
        TF  CC,*-90,11
PFONB   TR  C1&13,C2-26
        TF  MEMSV,REMAN,6
        TF  C1&73,REMAN
        AM  C1&73,1,10
        SM  REMAN,12,10
        TF  C1&78,REMAN
        AM  C1&78,1,10
        WNCDC1
        B   NBCTE&12
        DORG*-4
FLOT    TDM MEMSV,0,6
        SM  MEMSV,1,10
        BD  *&24,M-12
        BTM NBCTE,ZEL
        SF  M-12
        TF  CARA,DX,11
        SF  CARA-2
        MF  CARA,F9
        A   CARA,CC
        CM  CARA,100,9
        BNL SER&12
        CM  CARA,90,1011
        BNH SIMEL
        SF  F7
REAR    SF  CARA-1
        TF  CC,CARA
        TF  AR,CC
```

```
        SM   AR,10,10
        FADDCC,AR
        FSB  M,M-3
        BNF  SIMFL&56,F7
        BXV  SER&12
        CM   CC,1,10
        BH   PFONB
        BE   CTEFL
        CM   CC,1,1011
        BL   PFONB
        BH   SER&36
        C    M,UN-3
        BNF  PFONB
        BTM  NBCTE,CENMFL
SIMFL   CM   CARA,101,911
        BNH  FLOT&36
        CF   F7
        AM   CARA,10,10
        B    REAR
        DORG*-4
        CM   CC,90,1011
        BNH  FLOT&36
        SM   CC,10,10
        B    REAR&108
        DORG*-4
        DS   3
IMPER   RCTY
        DC   1,0,*-5
        TF   *-7,*-13
        WATYER
        WNTYIMPER&4
        RCTY
        AM   MEMSE,2,10
        WATYMEMSE,,6
        H
ER      DAC  4,ER 0,*-6
        B    BEGIN
        DORG*-4
        DS   5
NBCTE   TF   MEMSV,*-1,6
        AM   MEMSV,1,10
        B    EXIST2
        DORG*-4
SER     BTM  IMPER,002,9
        BTM  IMPER,001,9
        BTM  IMPER,003,9
        C    M,AR-5
        BH   PFONB
        BE   *672
        C    M,QUART
        BE   *660
        C    M,UN-3
        BNF  PFONB
        BTM  NBCTE,DIXMFL
        BTM  NBCTE,DEMIFL
        BTM  NBCTE,QUARFL
CTEFL   C    M,UN-3
        BE   *648
        C    M,DEUX
        BNF  PFONB
        BTM  NBCTE,DEUXFL
        BTM  NBCTE,UNFL
        MM   LTR&126,12,610
        SF   95
        A    99,ADRSUN
        BT   NBCTE,99
        BTM  NBCTE,ZEROFX
        BTM  NBCTE,DIXFX
        BTM  NBCTE,CENTFX
*
******     FONCTIONS USUELLES
*
FCTSTBTE  RESUL,MEMSV,11
        LD   99,RESUL
        SM   99,60473
        DM   95,5,10
        TF   FCNUM,97
        SM   MEMSV,5,10
        TD   C7,MEMSV,11
        CF   C7
        SF   FLF2
        CM   C7,1,10
        BE   *636
        BNF  STBES,FLR2
        CF   FLF2
        SM   MEMSV,1,10
        BNF  *624,MEMSV,11
        AM   MRINT,5,10
        MM   FCNUM,5,1011
        SM   99,FCTAB-55
        B    99,,6
        DORG*-4
FCAN    TDM  FCD,0,11
        BNF  FCT1,FLF2
        TR   C4,ZCOVR
        TF   C4&11,MEMSV,11
        BNF  *632,FL2RP
        TF   C4&18,MEMSV,11
        B    *632
        DORG*-4
        TF   C4&18,MRINT
        SM   MRINT,12,10
        BTM  FCSTOC,24,10
        B    FCT2
FCNUM   DS   ,*
FCT1    BNF  *620,FL2RP
        B    FCT2
        DORG*-4
        TR   C4,ZCOVR&12
        TF   C4&6,MRINT
        SM   MRINT,12,10
        TF   C4&11,MEMSV,11
        BTM  FCSTOC,12,10
FCT2    TR   C4,ZFCT
```

```
        TF   C4&11,MRINT
        AM   C4&11,12,10
        A    C4&6,RESUL
        SF   C4&6
        BTM  ECSTOC,12,10
        AM   MEMSV,6,10
        TF   MEMSV,C4&11,6
        AM   MEMSV,1,10
        TDM  MEMSV,0,611
ECD     DS   ,*
        TFM  *&30,ECEL
        A    *&18,ECNUM
        SF
        CM   ECNUM,11,10
        BNE  DERPAR&12
        AM   MRINT,11,10
        B    DERPAR&12
        DORG*-4
ECENT   BNE  *&116,FLE2
        BNE  *&24,FL2T
        SM   MRINT,5,10
        TF   *&35,MEMSV,11
        AM   MEMSV,6,10
        TFM  MEMSV,,6
        AM   MEMSV,1,10
        TD   MEMSV,C7,6
        ME   MEMSV,FL2RP,6
        B    DERPAR&12
        DORG*-4
        TDM  ECD,1,11
        B    ECT1
        DORG*-4
ECSG    TDM  ECD,1,11
        BNE  ECT1,FLE2
        BNE  *&20,FL2RP
        B    ECT2
        DORG*-4
        TR   C4,ZPREB1
        B    ECT1&32
        DORG*-4
ECABS   TD   ECD,C7
        SE   ECD
        B    ECSG&12
        DORG*-4
ECE     TDM  ECD,3,11
        B    ECSG&12
        DORG*-4
        DS   2
ECSTOCTR  MPGOB,C4,6
        A    MPGOB,*-13
        BB
        DORG*-10
ECTRI   SE   ECEL&19
        SE   ECEL&20
        B    ECAN
        DORG*-4
```

```
FLE2    DS   ,ECENT-1
        DC   5,50021
        DC   6,499950
ECEAD   DS   ,*-10
        DC   13,1631000000000,
ZECT    DS   ,*-12
ECTAB   DSA  ECE,ECABS,ECSG,ECENT,ECAN,ECAN,ECAN,ECAN,ECTRI,ECTRI
*
******     STOCKAGE DES CHAINES-FORMATS
*
GUIMETAM  MEMSO,2,10
        TFM  MEMSO,0,61011
        TF   RESUL,MEMSO
        TF   RAL,MEMSO
        AM   RAL,60,10
        SM   MEMSV,7,10
        TF   MONTAL&11,MEMSV
        SM   MONTAL&11,9,10
        TFL  FALX,XFL
        BTM  DECAL,13,10
        BE   FINAL
        CM   K,69,610
        BH   MULDYN&60
        TFM  *&35,TABAL-4
        A    *&23,4,10
        BNR  *&24
        BTM  IMPER,97,9
        C    *-13,K,611
        BNE  *-48
        AM   *-37,2,10
        TFM  *&30,'MULDYN '
        A    *&18,*-61
        B
        DORG*-4
MULDYNSE  EAL
RAL     DS   ,*
        AM   FALX,1,10
        TDM  FAL-1,1
        BTM  CALX,0,10
        BTM  MONTAL,GUIMET&96
        BT   CALX&24,CALX&23
        TDM  FAL-1,1
        B    *-36
        DORG*-4
        BTM  CALX,0,10
        BTM  MONTAL,*&12
        BTM  DECAL,13,1011
        BNE  *-24
        B    MULDYN&48
        DORG*-4
        DS   5
MONTALCM  RESUL
        BNH  *&24
        BTM  IMPER,98,9
        CM   FAL-2,396,9
        BNH  *&24
        BTM  IMPER,99,9
        AM   FAL-2,2,10
```

```
        AM   RESUL,2,10
        TF   RESUL,K,611
        B    *-100,,6
        DORG*-4
FINAL   BTM  CALX,0,10
        BTM  MONTAL,*&12
        BTM  MONTAL,*&12
        TFM  RESUL,,6
        DC   2,0,*
        BTM  DECAL,23,10
        BNE  *&32
        TFM  MEMSE,42,610
        B    *&56
        DORG*-4
        CM   K,4,610
        BE   *&24
        BTM  IMPER,95,9
        TFM  MEMSE,45,610
        BO   *&20,FAL-1
        B    *&32
        DORG*-4
        ME   FAL-1,FAL
        TD   MEMSO,FAL-1,6
        S    REMAN,FAL-2
        TF   MEMSV,REMAN,6
        AM   MEMSV,1,10
        TDM  MEMSV,7,6
        TF   ALMAX,REMAN
ALR     CM   FAL-2,60,10
        BNH  CHAIX
        TD   *&47,RAL,11
        TDM  RAL,,6
        DC   1,0,*
        BTM  ALPE,60,10
        TDM  RAL,,6
        SM   FAL-2,60,10
        TR   MEMSO,RAL,611
        SM   RAL,60,10
        B    ALR
        DORG*-4
        DS   5
ALPE    TR   C1,C2-30
        TR   C1,MEMSO,11
        TFM  C1&78
ALMAX   DS   ,*
        TF   C1&73,ALMAX
        A    C1&73,ALPE-1
        WNCDC1
        TF   ALMAX,C1&73
        BB
        DORG*-10
CHAIX   BT   ALPE,FAL-2
        SM   REMAN,1,10
        SM   MEMSO,2,10
        B    PRIOR
        DORG*-4
        DS   2
        BH   *&38
        CM   RESUL,67,610
        BE   *&14
        BB
        DORG*-10
        BTM  IMPER,96,9
FAL     DS   5
FALX    DS   2
        DC   5,0,
XFL     DC   2,0,
TABAL   DAC  1,0,
        DC   2,48,
        DAC  1,0,
        DC   2,48
        DAC  1,Z,
        DC   2,48,
        DAC  1,/,
        DC   2,48,
        DAC  1,@,
        DC   2,92,
        DAC  1,X,
        DC   2,0,
        DAC  1,N,
        DC   2,36,
        DAC  1,F,
        DC   2,36,
        DAC  1,E,
        DC   2,48,
        DAC  1,L,
        DC   2,36,
        DAC  1,-,
        DC   2,36,
        DAC  1,&,
        DC   2,36,
        DAC  1,.,
        DC   2,36,
        DAC  1,,,
        DC   2,36,
        DAC  1,%,
        DC   2,48,
        DAC  1,D,
        DC   2,36,
        DAC  1,S,
        DC   2,36,
        DAC  1,T,
        DC   2,36,
        DC   2,0,
```

```
*
******    COMPILATION DES PROCEDURES D E/S
*
PROCESTE  RESUL,MEMSV
          MM   CTSOR,6,9
          A    RESUL,99
          TF   NBVIR,99
          CM   RESUL,70000,67
          BL   *&36
          CM   RESUL,70003,67
          BL   *&108
          SM   RESUL,6,10
          CM   RESUL,70003,67
          BE   *&24
          BTM  IMPER,79,9
          SM   CTSOR,1,10
          C    FALX,CTSOR
          BE   *&60
          BTM  IMPER,93,9
          CM   FALX,0,10
          BE   *&24
          BTM  IMPER,92,9
          SM   RESUL,70000,67
          MM   RESUL,9,610
          AM   99,ESTAB
          TF   ESTR&23,99,11
          AM   99,2,10
          C    CTSOR,99,11
          BNL  *&24
          BTM  IMPER,88,9
          AM   99,2,10
          TF   ESCOL,99,11
          AM   RESUL,7,10
          TF   MEMSV,RESUL
          SM   RESUL,11,10
          TDM  RESUL,,6
          DC   1,0,*
ESJ       DS   ,*-1
ESJJ      DS   ,*-3
          TF   ESOR&30,RESUL
          TF   ESJ,CTSOR
          TFM  ESJJ,0,10
CONTES    SM   RESUL,1,10
          TD   C7,RESUL,11
          BNE  *&32,C7
          TFM  ESLIG,TABES1-34
          B    *&20
          DORG*-4
          TFM  ESLIG,TABES2-34
          AM   *&23,34
          BNR  *&24
ESLIG     DS   ,*
          BTM  IMPER,80,9
          C    *-13,C7,6
          BNE  *-48
          AM   ESLIG
ESCOL     DS   ,*
          TFM  *&30,STRES
          A    *&18,ESLIG,11
          B
          DORG*-4
          BTM  IMPER,81,9
          BTM  IMPER,82,9
          BTM  IMPER,83,9
          BTM  IMPER,84,9
          BTM  IMPER,85,9
          BTM  IMPER,86,9
          BTM  IMPER,87,9
          BTM  IMPER,89,9
STRES     BTM  IMPER,90,9
          BTM  IMPER,91,9
          BTM  IMPER,92,9
          BTM  IMPER,93,9
          AM   MRINT,1,10
          B    *&20
          DORG*-4
          AM   MRINT,12,10
          SM   RESUL,1,10
          BNE  *&24,RESUL,11
          AM   MRINT,5,10
          CM   ESJ,0,10
          BE   *&92
          CM   ESJJ,2,10
          BE   *&36
          AM   ESJJ,1,10
          AM   ESCOL,2,10
          SM   ESJ,1,10
          SM   RESUL,4,10
          B    CONTES
          DORG*-4
ESTR      TR   C4,ZAIG2&24
          TFM  C4&6
          TF   C4&11,MPGOR
          SM   C4&11,C3-18
          A    C4&11,CHOR
          AM   C4&11
NBVIR     DS   ,*
          BTM  ECSTOC,12,10
          SM   RESUL,4,10
          AM   NBVIR,6,10
          TFM  ESK,C3&150
          S    ESK,MPGOR
ESPE      C    ESK,NBVIR
          BH   ESOR
          TF   *&35,RESUL
          AM   *&23
ESK       DS   ,*
          TD   *&95
          TDM  *-1,,6
          DC   1,0,*
          TR   MPGOR,RESUL,611
          S    NBVIR,ESK
          A    MPGOR,ESK
          BTM  REQ,0,10
          A    RESUL,ESK
```

```
       TDM  RESUL,,6
       TFM  ESK,150
       B    ESPE
       DORG*-4
FSOR   TR   MPGOR,RESUL,611
       A    MPGOR,NBVIR
       TDM
       CM   ESTR&23,FORME
       BE   *&32
       TFM  CTSOR,0,10
       B    DEPPAR&12
       DORG*-4
       TFM  CTSOR,1,10
       TFM  FALX,0,10
       SM   MEMSV,6,10
       TF   MEMSV,PIFAD,6
       B    DEBPAR&12
       DORG*-4
ESTAB  DSA  SORTIR
       DC   2,0
       DC   2,2
       DSA  ENTRER
       DC   2,0,
       DC   2,10,
       DSA  SORTAB
       DC   2,2,
       DC   2,18,
       DSA  FORME
       DC   2,1,
       DC   2,26,
       DSA  PIF
PIFAD  DSC  1,7,
SORTIRDS    ,1111
ENTRERDS    ,2222
SORTABDS    ,3333
FORME  DS   ,4444
PIF    DS   ,55555
TABES1DC    2,-0
       DC   2,68
       DC   2,68
       DC   2,68
       DC   2,68
       DC   2,68
       DC   2,-84
       DC   2,-84
       DC   2,-84
       DC   2,68
       DC   2,-48
       DC   2,-36
       DC   2,-24
       DC   2,-12
       DC   2,&68
       DC   2,68
       DC   2,68
       DC   2,-01
       DC   2,68
       DC   2,68
       DC   2,68
       DC   2,68
       DC   2,68
       DC   2,-84
       DC   2,-84
       DC   2,-84
       DC   2,68
       DC   2,-48
       DC   2,-36
       DC   2,-24
       DC   2,-12
       DC   2,68
       DC   2,68
       DC   2,68
       DC   2,-03
       DC   2,-96
       DC   2,48
       DC   2,48
       DC   2,48
       DC   2,-96
       DC   2,-84
       DC   2,-84
       DC   2,-84
       DC   2,-96
       DC   2,-48
       DC   2,-36
       DC   2,-24
       DC   2,-12
       DC   2,12
       DC   2,12
       DC   2,12
       DC   2,-05
       DC   2,-96
       DC   2,-72
       DC   2,-72
       DC   2,-72
       DC   2,-96
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,-96
       DC   2,-48
       DC   2,80
       DC   2,-24
       DC   2,-12
       DC   2,12
       DC   2,12
       DC   2,12
       DC   2,-06
       DC   2,-96
       DC   2,-72
       DC   2,-72
       DC   2,-72
       DC   2,-96
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,-48
       DC   2,80
       DC   2,-24
       DC   2,-12
       DC   2,12
       DC   2,12
       DC   2,12
TABES2DC    2,0
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,-48
       DC   2,-36
       DC   2,-24
       DC   2,-12
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,1
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,-48
       DC   2,-36
       DC   2,-24
       DC   2,-12
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,3
       DC   2,-96
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,-96
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,-96
       DC   2,-48
       DC   2,-36
       DC   2,-24
       DC   2,-12
       DC   2,12
       DC   2,12
       DC   2,12
       DC   2,5
       DC   2,-96
       DC   2,-72
       DC   2,-72
       DC   2,-72
       DC   2,-96
       DC   2,80
       DC   2,80
       DC   2,80
       DC   2,-96
       DC   2,-48
       DC   2,80
       DC   2,-24
       DC   2,-12
       DC   2,12
       DC   2,12
       DC   2,12
       DC   2,7
       DC   2,-96
       DC   2,80
       DC   2,-60
       DC   2,-60
       DC   2,-96
       DC   2,80
       DC   2,-60
       DC   2,-60
       DC   2,-96
       DC   2,&80
       DC   2,-36
       DC   2,-24
       DC   2,80
       DC   2,12
       DC   2,12
       DC   2,12
       DC   2,9@,
```

```
******    PROGRAMME DE CHARGEMENT TRANSLATABLE SELECTIF
*
          DORG$1000
TRANS     TF   BOR,$200
          AM   BOR,5,10
          BNE  LECHAR,$REMAN$21
          TF   DEL,BOR
          SM   DEL,10000
          TEM  BOR0
FOR       DS   ,*-5
BOR       DS   ,*
          RD   CTOR,ZLOR$62
          TEM  BOK,ZLOR
          TEM  JOR,ZLOR$1
RETOR     AM   BOK,7,10
          CE   FLOR
          BNE  *$48
JOR       DS   ,*
          ME   TOR
BOK       DS   ,*
          AM   BOK,,6
DEL       DS   ,*
          ME   BOK,TOR,6
          BNE  MORO,FLOR
          AM   JOR,11,10
          BNR  RETOR,JOR,11
CTOR      A    ZLOR$60,DEL
          A    ZLOR$74,DEL
          TD   TOR,ZLOR$74,11
          TR   ZLOR$60,ZLOR$1,6
          TDM  ZLOR$74,0,6
TOR       DS   ,*
          TF   BOR,ZLOR$74
          RNCDZLOR$1
          CM   ZLOR$76,20,10
NEOR      DS   ,*
          BE   TRANS$32
BRSV      BNE  ECTSV,ZLOR$76
          B    DEMOR
          DORG*-4
MORO      AM   JOR,1,10
          AM   BOK,5,10
          SE   FLOR
          B    RETOR$24
          DORG*-4
FLOR      DS   1
ECTSV     SM   NEOR,1,10
          SM   FOR,5,10
          SM   TRANS$35,1,10
          B    TRANS$24
          DORG*-4
LECHAR    RNCDZLOR$1
          C    ZLOR$76,NEOR
          BE   *-24
          B    BRSV
          DORG*-4
ZLOR      DC   1,-1,50000
```



```
*
******    SOUS PROGRAMMES D EXECUTION
*
******    CONSTANTES COURANTES
*
TR        DS   ,588
M         DS   ,598
CC        DS   ,600
          DORG650
ZO        DC   24,0,
ZEROEXDS  ,ZO-12
UNEX      DC   12,1,
DEUXEXDC  12,2,
TROIEXDC  12,3,
QUATEXDC  12,4,
CINQEXDC  12,5,
DIXEX     DC   12,10,
CENTEXDC  12,100,
          DC   10,0,
ZEL       DC   2,-99,
          DC   10,5000000000,
DEMIELDC  2,00,
          DC   10,1000000000,
DIXMELDC  2,0,
          DC   10,1000000000,
CENMELDC  2,-1,
          DC   10,2500000000,
QUARELDC  2,0,
          DC   10,1000000000,
UNEL      DC   2,1,
          DC   10,2000000000,
DEUXELDC  2,1,
UNECEXDC  12,-1,
          DC   10,-1000001000,
UNECELDC  2,1,
          DC   10,9999999999,
UNEPS     DC   2,0,-
          DC   12,500000000000,
CQ        DC   2,0,
MAXEX     DC   12,999999999999,
          DC   10,9999999999,
MAXEL     DC   2,99,
*
******    SOUS PROGRAMME DE CONVERSIONS
*
          DORG980
          DS   12
CV        BNE  ENT,*-2
          TFL  CC,*-13
          FADDCC,DEMIEL
DE        CM   CC,12,10
          BH   ERN1
          TF   TR,ZEROEX
          CM   CC,1,1011
          BNH  FINCV
          BNE  *$24,M
          FSUBCC,UNEPS,10
          TEM  *$30,TR$1
```

```
        S   *&18,CC
        FSL TR&1=CC,M
        CE  *-6,,6
        SE  TR-11
        ME  TR,M
FINCV   BNE SORB&14,FLBB
        TE  CC,TR
        BR
        DORG*-10
FNT     CM  CV-1,0,10
        BE  AB&52
        TEM CC,13,10
        TEM *&47,CV-13,7
TN      AM  *&35,1,10
        SM  CC,1,10
        BD  *&20,CV-13
        B   TN
        DORG*-4
        TE  M,ZO-2
        ME  TN&43,CV-1
        TEM *&30,TR-2
        A   *&18,CC
        TE  TR-2&CC,CV-1
        SE  TR-1
        CM  CC,11,10
        BH  AB&20
        BE  AR
CG      FSR M,M-2
        ME  M,TN&43
SORB    BNE *&14,FLBB
        BR
        DORG*-10
        SE  FLBB
        R
        DORG*-4
FLBB    DC  1,-0,
AR      TEM CO,1,10
        B   *&20
        DORG*-4
        TEM CO,2,10
        FADDCC,CO
        B   CG
        DORG*-4
        TEL CC,ZEL
        B   SORB
        DORG*-4
ERA1    DAC 7,ER A1 @,
BUTER   DAC 8,ER A 11@,
        DORG1696
ERN1    TE  CC,MAXEX
        ME  CC,CV-3
        RCTY
        WATYERA1
        B   SORB

        DORG*-4
        DS  4
BUTOIRME  FL8,*-1
        SE  *-14
        TE  BUTER&6,*-27
        TNE BUTER&12,*-37
        RCTY
        WATYBUTER
        RCTY
        BNE *&20,FL8
BUTSORBB  ERVAR&20
        DORG*-4
        WATYBUT1
        H
        B   OUF
        DORG*-4
BUT1    DAC 21,EXECUTION IMPOSSIBLE@,
LBOUF   DAC 19,EXECUTION TERMINEE@,
OUF     CM  ZRCD,ZCD-2
        BE  *&24
        WACDZCD
        BNR *&68,ZTY
        RCTY
        RCTY
        WATYLBOUF
        H
        B   *-36
        DORG*-4
        RCTY
        WATYZTY
        B   *-80
        DORG*-4
R       DS  12
A       DS  12
        DS  5
AR      BV  *&12
        BNE BEIX,B-1
        BNE ABEX&12,A-1
        BD  *&100,A-11
        BD  *&44,B-11
        BTM BUTOIR,4102,811
        TEL A ,UNEL
        B   ABEX
        DORG*-4
        BNE ABEX,B-2
        BTM BUTOIR,4103,811
        TEL A,ZEL
        B   ABEX
        DORG*-4
        BNE *&36,A-2
        CE  A-2
        BTM BUTOIR,4104,811
        BD  *&20,B-11
        B   *-112
        DORG*-4
        BTM 553,A,67
        EMULA,B
        BTM 558,A,67

ABEX    BT  JNDAX,AB=1
        BT  CV,A
        TEL A,CC
        B   AB&36
        DORG*-4
FLAB    DS  1
BEIX    ME  *-1,B
        CM  B,1,10
        BE  BUN
        BL  BNUL
        BNE AENT,A-1
        BD  AEL,A-11
        BNE AB&116,FLAB
        B   AB&104
        DORG*-4
AEL     CM  A,1,10
        C   A-2,UNEGEL-2
        BNE *&60
        BE  PARITE
        C   A-2,UNEL-2
        BE  ABEX
        CM  B,30,10
        BH  RCONV
        SM  B,1,10
        TEL TR,A
        EMULA,TR
        BV  DDC
        SM  B,1,10
        BH  *-26
RECONVBNE ABEX,FLAB
        TEL TR,UNEL
        EDIVTR,A
        TEL A,TR
        B   ABEX
        DORG*-4
PARITELD  99,B
        DM  BB,2,10
        CM  BB,0,10
        BE  AB&72
        TEL A,UNEGEL
        B   ABEX
        DORG*-4
DDC     BE  BEIX&72
        BTM BUTOIR,4105,811
        TEL A,MAXEL
        B   ABEX
        DORG*-4
AENT    CM  A,0,10
        BE  BEIX&72
        CM  A,1,1011
        BE  PARITE
        CM  A,1,10
        BE  AB&72
        BNE *&44,FLAB
        BT  CV,A
        TEL A,CC

        B   AEL&72
        DORG*-4
        SM  B,1,10
        TE  TR,A
        TE  80,ZO
        M   A,TR
        CM  87,0
        BNE DDC&12
        SE  88
        TE  A,99
        SM  B,1,10
        BH  *-94
CONV    BT  CV,A
        TEL A,CC
        B   RECONV
        DORG*-4
BNUL    BNE *&32,A-1
        BD  AB&72,A-11
        B   AB&60
        DORG*-4
        CM  A,0,10
        BNE AB&72
        B   AB&60
        DORG*-4
BUN     BNE CONV,A-1
        B   RECONV
        DORG*-4
RCONV   BT  CV,B
        TEL B,CC
        B   AB&36
        DORG*-4
```

```
*
******     PROCEDURES USUELLES SOUS/PROGRAMMES FIXES
*
ENDD  TF  FAC,SAVE
      BNE *&24,99
      SF  FAC-2,ENDD
FINISHDS  ,*
      TFL ,FAC,6
OUT   DS  ,*-5
      BR
      DORG*-10
OVFLO TF  FAC-2,UNFRS-2
      TFM FAC,99,10
      B   ERXV&24
      DORG*-4
ZFREACTFL OUT,ZFL,6
      BR
      DORG*-4
UNFLO TFL FAC,ZFL
      B   ERXV&12
OVFLOWTFL FAC,MAXNEG
ERROR RCTYFINISH
      WATYMESERR
      B   ENDD&12
      DORG*-4
ERXV  BNE *&24,FAC
      AM  FI,1,10
      RCTYENDD&36
      WATYMESERR
      B   ERXV&30,,6
NORM  DS  ,*-39
      DORG*-4
      TR  FAC-11,FAC-10
      TDM FAC-1,0
      SF  FAC-11
      BD  FINISH,FAC-11,6
      SM  SAVE,1,10
      BNV NORM&36
      TFM FI,572,9
UNFLOWTFL FAC,ZFL
      B   ERROR
      DORG*-4
MU    DS  ,ERXV
INDES TF  OUT,ADES,11
      TFM TRAD-1,OUT
      TDM BB,9
      TFM BB&5,*&20
      B   TRAD
      DORG*-4
      TDM BB,2
      AM  ADES,21,10
      B   ADES&21
ADES  DS  ,*-5
      DORG*-4
      DC  13,0
9SPF  DS  ,*-2

LOG2  DC  13,0693147180560,
LOG4  DC  13,1386294361120,
LOG8  DC  13,2079441541680,
LOG10 DC  13,2302585092994,
      DC  14,10000000000000,
ONEZ  DS  ,*-4
      DC  2,01
PIOV2 DC  12,157079632679,
      DC  10,-9999999999,
MAXNEGDC  2,99,
      DC  40,0
      DSC 40,0
ZERO  DS  ,*
MESERRDAC 6,ER F @,
FI    DS  ,*-2
FAC   DS  60
SAVE  DS  73
RETA  DS  38
      DC  1,@,
IMSA  DS  ,RETA-28
      DC  1,@,FAC&1
```



```
*
*****     SS/PR DE DEPART DU PROGRAMME
*
REMAN DS  ,49999
BORO  DS  ,573
TABLOCDC  35,00003999984000284998900014000340019,
RECHARDAC 20,RECHARGER EXECUTION@,
DAPOR TFM *&23,REMAN&21
      BNE *&56
      RCTY
      WATYRECHAR
      H
      B   *-36
      DORG*-4
      CM  DAPOR&23,REMAN&1
      BE  *&32
      SM  DAPOR&23,1,10
      B   DAPOR&12
      DORG*-4
      TF  BOR,BORO
      AM  BOR,5,10
DENOR TF  TABLO,TABLOC
      TFM *&18,TABLO
      A   ,BOR
      SM  *-6,5,10
      CM  *-18,TABLO-35
      BNE *-36
      TF  LIMIN,TABLO-30
      TFM TABLO-20,0,67
      TF  REMAN-5,REMAN
      SM  REMAN-5,5000
      TF  REMAN-10,REMAN-5
      AM  REMAN-10,5,10
      TFM REMAN-15,0,10
      TFM ZRCD,ZCD-2
      TFM ZRTY,ZTY-2
      TFM *&18,ZTY
      TFM
      DC  2,@,*
      AM  *-6,2,10
      CM  *-18,ZTY&162
      BNE *-36
      H
      B   20000
      DORG*-4
```

```
*
******    EXECUTION PROCEDURE ENTRER
*
         DS   5
ENTRER   TEM  FL1,0,10
         PT   ESEIN&1,*-13
         BTM  ESTYP,0,10
         BTM  FLSV,ENTITR
         CM   DT,3,10
         BH   TAB
         BE   EBOOL
         BO   NBENT,DT
         SE   F2
NB       TR   F6,FLAG
         TEM  CH&114,M-13
         TEM  EXP&162,CARA-3
         TEM  CC,0,10
         TE   M,ZO
         SE   F7
FL1      DS   ,*
FL2      DS   ,*-1
FL3      DS   ,*-2
FL5      DS   ,*-3
FL4      DS   ,*-4
         BTM  DK,0,10
         BE   *-12
         CM   K,20,610
         BNE  CH&132
         SE   F8
F1       DS   ,*
F2       DS   ,*-1
F3       DS   ,*-2
F7       DS   ,*-3
CH       BTM  DK,70,10
         BL   PT
         BNE  CHD,F7
         BNE  CH&176,F5
         AM   CC,1,10
         BNE  CH,F6
         CM   *&42,M-1
         BE   CH&19*12-8
         AM   *&18,1,10
         TD   ,K,11
         B    CH
         CM   K,10,610
         BE   CH
         CM   K,70,610
         B    CH&12
         DORG*-4
         SE   F4
         BE   CH
         SE   F5
         B    CH&48
         DORG*-4
         CE   F6
FL6      DS   ,*
FL7      DS   ,*-1
FL8      DS   ,*-2
FL9      DS   ,*-3
FL10     DS   ,*-4
         B    CH&96
         DORG*-4
PT       CM   K,3,610
         BNE  EXP-12
         BTM  DK,70,10
         BL   ZEXP
         BNE  FRL4,F7
         CE   F7
         SE   F4
F14      DS   ,*-4
F13      DS   ,*-3
F12      DS   ,*-2
F10      DS   ,*
CHD      BNE  *&20,F5
         B    CH&60
         DORG*-4
         BE   *&32
         SE   F5
         B    CH&96
         DORG*-4
         SM   CC,1,10
         B    CH
         DORG*-4
         CM   K,45,610
EXP      BNE  VID
         CE   F7
         BNE  EXPUR,F4
         CE   F5
         BTM  DK,20,10
         BE   EXPUR&48
         CM   K,10,610
         BE   EXPUR&60
         CM   K,70,610
         BL   FRL4
         BE   *&48
         SE   F5
         AM   *&18,1,10
         TD   CARA-3,K,11
         BTM  DK,70,10
         BL   VID&12
         BNE  EXPUR&160,F5
         CM   *-42,CARA
         BNE  EXP&144
FRL3     TEM  AUTOIR-1,4503,811
         TEM  ERVAR&154,RELEC
         B    ERVAR
         DORG*-4
FRL4     TEM  AUTOIR-1,4504,811
         B    *-32
         DORG*-4
EXPUR    TEM  CC,1,10
         TE   M,UN13
         BTM  DK,20,10
         BNE  *&44
         SE   F8
         BTM  DK,70,10
         B    EXP&108
         DORG*-4
         CM   K,10,610
         BE   EXPUR&60
         CM   K,70,610
         BH   EXP&132
         BE   EXP&168
         BNE  ENTRAL,F7
         B    FRL4
         DORG*-4
         BE   EXP&168
         B    EXP&132
         DORG*-4
VID      BNE  ENTRAL,F4
         CM   K,0,610
         BNE  FRL4
         SE   M-12
         TE   CARA,EXP&162,11
         SE   CARA-2
         ME   CARA,F8
         A    CARA,CC
         BNE  ENTS,F2
         BO   *&32,M-12
PLNUL    TEL  SCS,ZEL
         B    FINNB
         DORG*-4
         CM   CARA,100,9
         BNL  FRL3
         CM   CARA,90,1011
         BNH  RL
         SE   F7
REA      SE   CARA-1
D        DS   ,*
         TE   CC,CARA
         TE   AR13,CC
         SM   AR13,10,10
         EADDCC,AR13
         BXV  FRL3
         BNE  ENTS&68,F2
         ESR  M,M-2
         BNE  PL&56,F7
         ME   M,F8
         TEL  SCS,CC,6
FINNB    BNE  *&32,FL4
         BT   FLSV,ESEIN
         B    ENTRER&48
         DORG*-4
         CM   SCS
         BE   *-32
         AM   SCS
         B
         DORG*-4
RL       CM   CARA,101,911
         BNH  PLNUL
         CE   F7
         AM   CARA,10,10
         B    REA
         DORG*-4
         CM   CC,90,1011
         BNH  PLNUL
         SM   CC,10,10
         B    REA&108
         DORG*-4
ENTS     CM   M,0,10
         BE   *&36
         CM   CARA,1,1011
         BH   PLNUL&20
         TE   SCS,ZEROEX,6
         B    FINNB
         DORG*-4
         TEM  TR-11,0,9
         TEM  AR13,00,10
         EADDCC,AR13
         CM   CC,0,10
         BNH  ENTS&48
         CM   CC,12,10
         BH   FRL3
         TEM  *&30,TR-1
         S    *&18,CC
         ESL  TR-1-CC,M
         CE   *-6,,6
         SE   TR-13
         ME   TR-2,F8
         TE   SCS,TR-2,6
         B    FINNB
         DORG*-4
ENTRAL   BTM  DK,0,10
         BNE  *-12
         B    NB&36
         DORG*-4
ZEXP     CM   K,69,610
         BL   FRL4
         SE   F7
         B    EXP&24
         DORG*-4
NBENT    CE   F2
ADT      DS   ,*
         B    NB
         DORG*-4
TAB      CM   DT,5,10
         BNE  TABOOL
         ME   F2,FL6
         TEM  FINNB&74,NB
         TEM  FINNB&67,12,10
         TE   ADT,SCS
         TE   SCS,SCS,11
         SM   ADT,5,10
         MM   ADT,7,610
         S    ADT,99
```

```
       SM   ADT,3,10
       TF   D,ADT,11
       SM   ADT,8,10
       TF   FINNB&43,ADT
       SM   D,1,10
       BF   *&68
       SM   ADT,10,10
       M    ADT,FINNB&43,6
       SF   95
       TF   FINNB&43,99
       B    *-72
       DORG*-4
       M    FINNB&43,FINNB&67
       SF   95
       TF   FINNB&43,99
       A    FINNB&43,SCS
       S    FINNB&43,FINNB&67
       B    FINNB&74,,6
       DORG*-4
TABOOLTFM FINNB&74,FBOOL
       TFM FINNB&67,1,10
       B    TAB&60
       DORG*-4
FBOOL  BTM DK,0,10
       BF   *-12
       CM   K,65,&10
       BNF  FFAUX
       TDM *&47,0
       BTM DK,0,10
       BNF  FFAUX&44
       TDM SCS,0,6
       B    FINNB
       DORG*-4
FFAUX  CM   K,46,&10
       BNF  FBOCH
       TDM FBOOL&95,1
       B    FBOOL&60
       DORG*-4
       BD   *&116,FBOOL&95
       CM   K,59,&10
       BNF  FRL2
       BTM DK,41,10
       BNF  FRL2
       BTM DK,49,10
       BNF  FRL2
       BTM DK,0,10
       BF   FBOOL&84
       B    FRL2
       DORG*-4
       CM   K,41,&10
       BNF  FRL2
       BTM DK,64,10
       BNF  FRL2
       BTM DK,67,10
       BF   *-92
FRL2   TFM BUTOIR,4602,811
       B    FRL3&12
       DORG*-4
FBOCH  CM   K,7,&10
       BL   *&92
       CM   K,71,&10
       BH   *&68
       BTM CHANBO
K      DS   ,*
       TD   FBOOL&95,CHANBO-1
       BTM DK,0,10
       B    FBOOL&84
       BNF  FRL2
       DORG*-4
       BTM DK,0,10
       BNF  *-12
       B    FBOOL
       DORG*-4
       DS   5
CHANBORD  *&26,*-1,11
       TDM *-13,1
       BB
       DORG*-10
       TDM *-27,0
       BB
       DORG*-10
FNTITRBTM DK,0,10
       TR   PIE&1,ZIN-1
       TDM PIE,2
       B    FSEIN,,6
       DORG*-4
SORAL  CM   PIE,2,10
       BNF  *&44
       BTM AVZS,160,711
       TR   SZS-1,PIE&1,6
       B    FSEIN,,6
       DORG*-4
       BTM BUTOIR,6208,811
       B    FSEIN,,6
       DORG*-4
       DS   2
DK     AM   K,2,10
       CM   K,ZIN&160
       BF   LEC
       CM   K,0,&10
       BF   *&38
       SF   FL3
       C    K,DK-1,6
       BB
       DORG*-10
       BNF  DK&60,FL3
       CF   FL3
       B    DK,,5
       DORG*-4
LEC    BNR  *&88,ZIN&160
       TR   ZIN-1,PLANCS-1
       BNF  *&44,FL2
       RACDZIN
       TFM K,ZIN
```



```
       B    DK&36
       DORG*-4
       RATYZIN
       B    *-32
       DORG*-4
       BNR  *&116,BLANCS&160
       TR   ZIN&159,BLANCS&159
       TFM *&18,BLANCS
       TFM BLANCS,0,10
       AM   *-6,2,10
       CM   *-18,BLANCS&160
       BL   *-36
       TFM BUTOIR-1,4505,811
       TFM ERVAR&166,LEC&12
       B    ERVAR
       DORG*-4
       BTM BUTOIR,4506,8
ERVAR  TDM BUTSOR&1,9
       B    BUTOIR
       DORG*-4
       TDM BUTSOR&1,2
       WATYZIN
       RCTY
       TF   *-6,K
       SM   *-18,ZIN
       CM   *-30,0,10
       BF   *&44
       SPTY
       SM   *-66,2,10
       B    *-48
       DORG*-4
       WATYSTAR
       B
       DORG*-4
RFLEC  RCTY
       WATYCOL
       TFM *&74
       RNTY*&61
       BNR  *&20,*&40
       B    ENTRER
       DORG*-4
       A    *&30,*&30
       AM   *&18,ZIN-2
       RATY
       TF   K,*-6
       SF   FL3
       BNF  FINNB&74,FL4,6
       B    ENTRER&48
       DORG*-4
COL    DAC 17,NUMERO COLONNE* @,
STAR   DAC 2,*@,
*
******     EXECUTION PROCEDURE SORTIR
*
       DS   5
SORTIRTFM FL1,0,1011
       BT   FSEIN&1,*-13
       BTM FSTYP,0,10
       BTM ELSV,SORAL
       BNF  PARA2&12,F5
       BD   FORSTA,SCS,11
       BNF  PARA2,SCS,11
       TF   SF,SCS
       NOP
APEL1  TFM LONG1
       AM   SF,2,10
       CM   SF,21,&10
       BNF  *&32
       BTM LIGNE,0,10
       B    *-48
       DORG*-4
       CM   SF,62,&10
       BNF  *&32
       BTM SAUTY,0,10
       B    *-92
       DORG*-4
       CM   SF,13,&10
       BF   FINS
       SM   SF,2,10
       TF   SF2,SF
       BTM CTEOR,5,10
       DS   2
ZONE1  AM   LONG1,2,10
       BB
       DORG*-8
ZONEN  BTM ELSV,FRSO2
       AM   LONG1,2,10
       CM   DT,3,10
       BF   CTEOR
       AM   LONG1,28,10
       B    CTEOR
       DORG*-4
       DS   2
VIRG1  TFM SF
SF2    DS   ,*
       CM   LONG1,160,9
       BNH  *&24
       BTM SECOUR,6201,811
       BTM AVZS
LONG1  DS   ,*
       BTM CTEOR,10,10
PARA2  BTM FLSV,FINS
       BTM FORNOR,*&12
       BTM AVZS,4,711
       TFM SZS,0,68
```

```
       B   *-48
       DORG*-4
       DS  5
FLSV   SM  SCT,11,10
       BNE *-13,SCT,611
       AM  SCT,5,10
       TEM *&35
SCT    DS  ,*
       SM  *&23,1,10
       TE  SCS
       TEM TRAD-1,SCS
       TDM BB,9
       TEM BB&5,*&20
       B   TRAD
       DORG*-4
       TDM BB,2
       TD  DT,SCT,11
       ME  FL6,DT
       TEM FL5,0,10
       CM  DT,4,10
       BH  *&14
       BB
       DORG*-4
       CM  DT,7,10
       BE  *&26
       CE  FL4
       BB
       DORG*-10
       SE  FL5
BAE    BB
       DORG*-10
       DS  2
CTEOR  AM  SE,2,10
       CM  SE,60,610
       BH  REPLI
       TEM *&30,FORTAB-32
       AM  *&18,32,10
       C   ,SE,11
       BNE *-24
       A   *-18,*-85
       SE  *-30
SE     DS  ,*
       BTM *-42,0,610
       B   *-120
       DORG*-4
       DS  2
FSTYP  BD  *&80,SCT,11
       TEL CC
SCS    DS  ,*
       CM  CC,1,10
       BNE FSCD
       C   CC-2,UNFL-2
       BNE FSCD
       B   FSTY
       DORG*-4
       TE  CC,SCS,11
       CM  CC,1,10
       BE  FSTY
FSCD   BNE *&38,FL1
       TEM *&24
ZRCD   DS  ,*
       TEM BORZS,ZCD&158
       BB
       DORG*-10
       TEM SZS,ZIN-1
       TR  ZIN-1,BLANCS-1
       SE  FL3
       TEM K,ZIN&158
       BB
       DORG*-10
FSTY   CE  FL2
       BNE FSCD&38,FL1
       TEM SZS
ZRTY   DS  ,*
       TEM BORZS,ZTY&158
       BB
       DORG*-10
FSEIN  DS  6
       TE  SCT,*-1
       SM  SCT,1,10
       B   FLSV&36
       DORG*-4
FINS   BNE *&32,FL2
       TE  ZRCD,SZS
       B   FSEIN,,6
       DORG*-4
       TE  ZRTY,SZS
       B   FSEIN,,6
       DORG*-4
       DS  2
MAR1   BNE RETAB,FL4
       BTM FLSV,FINS
       BTM BUTOIR,6203,811
       B   PARA2&12
       DORG*-4
       DS  2
AR41   AM  SE,2,10
       CM  SE,13,610
       BE  CTEOR
       AM  LONG1,2,10
       B   *-48
       DORG*-4
       DS  5
AVZS   ME  FL7,*-1
       AM  *-13
SZS    DS  ,*
       CM  *-25
BORZS  DS  ,*
       BH  *&38
       BNE *&24,FL7
       TE  SZS,AVZS-1
       BB
       DORG*-10
       TDM LIGNE&40,0
       B   LIGNE
       DORG*-4
AVZS1  TDM LIGNE&40,2
       SM  AVZS-1,160,9
       B   AVZS&48
       DORG*-4
       DS  2
LIGNE  BNE *&56,FL2
       MACDZCD
       TR  ZCD-1,BLANCS-1
       TEM SZS,ZCD-2
       BB  AVZS1
       DORG*-4
       BNE *&44,ZTY
       TDM *&23,2
       RCTY
       B   *-44
       DORG*-4
       TDM *-9,2
       WATYZTY
       TEM *&18,ZTY
       TEM
       DC  2,0,*
       AM  *-6,2,10
       CM  *-18,ZTY&162
       BNE *-36
       TEM SZS,ZTY-2
       B   *-116
       DORG*-4
       DS  2
SAUTY  BNE *&14,FL2
       BB
       DORG*-10
       BNE *&26,ZTY
       TRTY
       BB
       DORG*-10
       TDM LIGNE&91
       B   LIGNE&112
       DORG*-4
       DS  2
ARABESAM   SE,2,10
       CM  SE,13,610
       BE  CTEOR
       BTM AVZS,2,711
       TE  SZS,SE,611
       B   *-60
       DORG*-4
       DS  2
ESPACEBTM  AVZS,2,711
       TEM SZS,0,610
       B   CTEOR
       DORG*-4
       DS  2
NOPMA  BTM FORNOR,ABFL4
       DS  5
FORNORBNE  NORTAB,FL4
       CM  DT,3,10
       BE  SORBO
       TR  STR-1,BLANCS&127
       BD  SORN,DT
       BTM AVZS,32,711
       TEL CC,SCS,11
       TEM STR&2,3,10
       BD  *&32,CC-11
       TEM STR&4,70,10
       B   *&116
       DORG*-4
       BNE *&36,CC-2
       CE  CC-2
       TEM STR,20,10
       BNE STR&22,CC-2
       TEM STR&24,45,10
       BNE *&36,CC
       TEM STR&26,20,10
       CE  CC
       BNE STR&30,CC
       BTM STOCNO,STR
SORN   BTM AVZS,26,711
       TE  CC,SCS,11
       CM  CC,0,10
       BE  *&200
       TEM *&35,CC-11
       TEM *&110,STR&8
       BD  *&44
       AM  *-1,1,10
       AM  *&74,2,10
       B   *-36
       DORG*-4
       CM  *-33,CC
       BE  *&84
       SE  *-57,,6
       BNE *&26,CC
       TEM STR&8,20,10
       CE  CC
       BNE STR&30,CC
       BTM STOCNO,STR&6
       BNE *&24,CC
       TEM STR&28,20,10
       TD  STR&30,CC
       TDM STR&29,7
       B   *-60
       DORG*-4
       DS  5
STOCNOCE   *-1,,6
       AM  *-13,1,10
       CM  *-25,STR&31
       BNE STOCNO
       TE  SZS,STR&30,6
       B   FORNOR-1,,6
       DORG*-4
SORBO  BTM AVZS,2,711
       BD  *&32,SCS,11
       TEM SZS,65,610
       B   FORNOR-1,,6
       DORG*-4
       TEM SZS,46,610
```

```
        B   FORNOR-1,,6
        DORG*-4
        DS  2
LOGIO   BNE *&48,FL4
        BTM FLSV,FRS02
        CM  DT,3,10
        BNE FRS04
        BTM AVZS,2,711
        TEM SZS,70,610
        BT  CHANRO,SCS
        TD  SZS,CHANRO-1,6
        BTM CTEOR,20,10
        DS  2
BOOL    BNE *&48,FL4
        CM  DT,3,10
        BNE FRS04
        AM  SE,2,10
        CM  SE,46,610
        BE  *&24
        BTM FORNOR,APFL4
        AM  SE,2,10
        CM  SE,46,610
        BE  *-24
        BTM AVZS,8,711
        BD  *&32,SCS,11
        TF  SZS,VRAI,6
        B   APFL4
        DORG*-4
        TF  SZS,FAUX,6
        B   APFL4
        DORG*-4
        DS  6
DC3     AM  SE,2,10
        CM  SE,42,610
        BE  *-24
        CM  SE,34,610
        BE  *&20
        B   DC3-1,,6
        DORG*-4
        AM  SE,2,10
        CM  SE,13,610
        BNE *-24
        B   DC3
        DORG*-4
        DS  2
NOMBRESM SE,2,10
        TF  SE2,SE
        TR  L1-1,BLANCS&147
        TF  ELP2,Z0-16
        BTM DC3,*&12
        CM  SE,20,610
        BE  APSG&20
        CM  SE,10,610
        BE  APSG
        CM  SE,69,610
        BNE *&36
        AM  L1,1,10
        BTM DC3,*-36
        CM  SE,44,610
        BNE *&36
        AM  L2,1,10
        BTM DC3,*-36
        CM  SE,63,610
        BE  APT1
        CM  SE,3,610
        BE  APT1&36
        CM  SE,45,610
        BNE APNB
        SE  ELE
        BTM DC3,*&12
        CM  SE,10,610
        BE  APT1&18*12
        CM  SE,20,610
        BE  APT1&236
APZE    CM  SE,69,610
        BNE *&108
        AM  L4,1,10
        BTM DC3,*&12
        CM  SE,69,610
        BE  *-36
        CM  SE,44,610
        BNE APNB
        AM  L5,1,10
        BTM DC3,*-36
        CM  SE,44,610
        BE  *-36
        BTM SECOUR,6210,811
APSG    SE  ELP1
        B   *&20
        DORG*-4
        SE  ELM1
        BTM DC3,*&12
        CM  SE,69,610
        BE  NOMBRE&132
        CM  SE,44,610
        BE  NOMBRE&180
        CM  SE,63,610
        BE  APT1
        CM  SE,3,610
        BE  APT1&36
        CM  SE,45,610
        BE  NOMBRE&23*12
        BTM SECOUR,6208,811
APT1    AM  L2,1,10
        SE  ELT
        BTM DC3,NOMBRE&21*12
        BTM DC3,*&12
        CM  SE,44,610
        BNE *&120
        AM  L3,1,10
        BTM DC3,*&12
        CM  SE,44,610
        BE  *-36
        CM  SE,63,610
        BNE NOMBRE&21*12
        SE  ELT
        AM  L3,1,10
        BTM DC3,NOMBRE&21*12
        CM  SE,63,610
        BE  *-48
        BTM SECOUR,6209,811
        SE  ELP2
        B   *&20
        DORG*-4
        SE  ELM2
        BTM DC3,APZE
CONEORBT CV,SCS,11
        B   APNB&84
        DORG*-4
        TF  AR10,L3
        FADDCC,AR10
        B   APEX
        DORG*-4
APNB    BNE *&84,FL4
        BTM FLSV,FRS02
        CM  DT,1,10
        BNH *&24
        BTM SECOUS,6205,811
        BE  CONEOR
        TEL CC,SCS,11
        BD  *&44,M-9
        TF  CC,L3
        SE  CC
        B   *&32
        DORG*-4
        ME  FL8,CC-2
        BNE CONEOR&20,FLT
APEX    BNE *&24,FLE
        BD  PREX,M-9
        TF  *&35,L1
        A   *&23,L2
        CM  CC
        BNH *&24
        BTM SECOUS,6206,811
        TF  NBZ,L2
        S   NBZ,CC
        BL  *&84
        TF  NBB,L1
        TEM POSD,M-10
        TEM BPOSD,M
        TF  SE,SE2
        TEM DERZ,0,10
        BTM CTEOR,15,10
        TF  NBB,NBZ
        A   NBB,L1
        TEM NBZ,0,10
        B   *-96
        DORG*-4
PREX    TF  NBZ,L1
        A   NBZ,L2
        TF  NBB,DIXEX
        S   NBB,L3
        TF  SAVE,ZEROEX
        A   SAVE,CC
        C   NBZ,NBB
        BH  *&44
        S   SAVE,NBZ
        TF  CC,NBZ
        B   APEX&24
        DORG*-4
        TF  NBZ,NBB
        B   *-44
        DORG*-4
        DS  2
PLUS3   BNE SIGB,FLB1
        BNE *&20,FL8
        B   MOINS3&12
        DORG*-4
        TEM SIG,10,10
        B   MOINS3&24
        DORG*-4
        DS  2
MOINS3  BNE SIGB,FL8
        TEM SIG,20,10
        CM  NBB,0,10
        BH  *&44
        BTM AVZS,2,711
        TF  SZS,SIG,6
        B   CTEOR
        DORG*-4
        TF  DERZ,SIG
        B   ESPACE
        DORG*-4
SIGB    TEM SIG,0,10
        B   MOINS3&24
        DORG*-4
        DS  2
ZER3    SM  NBB,1,10
        BH  ESPACE
        BNE TRDIG
        BTM AVZS,2,711
        TF  SZS,DERZ,6
        B   CTEOR
        DORG*-4
        DS  2
DIG3    SM  NBZ,1,10
        BL  *&44
        BTM AVZS,2,711
        TEM SZS,70,610
        B   CTEOR
        DORG*-4
TRDIG   BTM AVZS,2,711
        TEM SZS,70,610
        AM  POSD,1,10
        CM  POSD
BPOSD   DS  ,*
        BH  ESPACE
        TD  SZS,,6
POSD    DS  ,*
        B   CTEOR
```

```
DORG*-4
DS   2
BTM  AVZS,2,711
TFM  SZS,3,610
B    CTEOR
DORG*-4
DS   2
BTM  AVZS,2,711
TFM  SZS,45,610
BT   CV,SAVE
BO   *&24,M-9
TF   CC,L5
TR   L1-1,L3-1
TR   FLM1,FLM2
TF   SE2,SE
B    APEX&24
DORG*-4
BTM  BUTOIR,6202,811
B    FINS
DORG*-4
DS   2
SM   SE,2,10
BTM  CTEOR,20,10
DS   4
BNE  FRELTA,FL4
BT   BUTOIR,*-13
B    PARA2
DORG*-4
DS   4
BNE  FRELTA,FL4
BT   BUTOIR,*-13
B    PARA2&12
DORG*-4
BTM  BUTOIR,6204,811
BTM  FORNOR,APFL4
DS   2
BTM  BUTOIR,6207,811
AM   SE,2,10
CM   SE,13,610
BE   MAR1
CM   SE,23,610
BE   APEL1
B    *-60
DORG*-4
*
******     EXECUTION PROCEDURE FORMAT
*
         DS   6
DEMUL    ME   FL9,SCS,11
         TF   SE,SCS,
         TFM  SPO,P0&5
         TFM  SPV,PIE&2
         BTM  CTEOR,25,10
REPLI    TO   REP-1,SE,11
         SF   REP-1
         AM   SE,2,10
SPO      DS   ,*
         CM   SE,69,610
         BNH  *&44
         SM   SE,2,10
         TFM  ASPO,REP-1
         B    MONTPO
         DORG*-4
         TO   REP,SE,11
         SE   REP
SPV      DS   ,*
         TFM  ASPO,REP
MONTPO   BNE  *&24,SPO,11
         BTM  ERMUL,6212,811
         SM   SPO,5,10
         TF   SPO,,6
ASPO     DS   ,*
         B    CTEOR
         DORG*-4
         DS   2
FORX     BTM  FLSV
         TF   CC,SCS,11
         BO   *&24,DT
         BTFLCV,SCS,11
         CM   CC,1,1011
         BH   *&24
         BTM  BUTOIR,6213,811
         CM   CC-2,0,10
         BE   *&24
         BTM  BUTOIR,6214,811
         SF   CC-1
REP      DC   3,0,*
         SF   CC
         TFM  ASPO,CC
         B    MONTPO
         DORG*-4
         DS   2
PARO     BNE  *&44,SPO,11
         SM   SPO,2,10
         TF   SPO,SPV,6
         B    CTEOR
         DORG*-4
         SM   SPO,5,10
         B    *-32
         DORG*-4
         DS   2
ARAS     BTM  MONTV,SE,711
         CM   SE,13,610
         BE   CAR5&12
         AM   SE,2,10
         B    *-48
         DORG*-4
         DS   5
MONTV    TF   SPV,*-1,611
         AM   SPV,2,10
         CM   SPV,MAXV
         BH   PAF
         BTM  ERMUL1,6215,811
         DS   2
CAR5     BTM  MONTV,SE,711
         BNE  CTEOR,SPO,11
         TF   SPP,SPV
         SM   SPP,3,10
         B    MULEE
         DORG*-4
         DS   2
PARF     SM   SPV,2,10
         CM   SPV,23,610
         BE   *&24
         BTM  MONTV,VIRSUP
         AM   SPV,2,10
         BNE  *&24,SPO,11
         BTM  ERMUL,6216,811
         TF   SPP,SPO,11
         AM   SPO,2,10
         BNE  XNUL&20,SPO,11
         SM   SPP,1,10
MULEE    TF   SPV1,SPV
         TFM  SPV,,6
         DC   2,0,*
         AM   SPV1,1,10
VIRSUP   AC   1,,,*-2
         TF   SPVP1,SPV1
         TF   SPVM1,SPV
         SM   SPVM1,1,10
         TF   LM,SPVM1
         SM   LM
SPP      DS   ,*
         TF   SPP1,SPO
         AM   SPP1,1,10
         CM   SPO,0,610
         BE   XNUL
         CF   SPO,,6
         MM   SPO,,6
LM       DS   ,*
         A    99,SPP1
         CM   99,MAXV
         BNH  *&24
         BTM  ERMUL1,6215,811
         SF   05
         TF   SPV,99
         SM   SPO,1,610
         BE   *&44
         TR   ,SPP,11
SPV1     DS   ,*-5
         A    *-6,LM
         B    *-48
         DORG*-4
         TR
SPVP1    DS   ,*
SPVM1    DS   ,*-5
         AM   SPO,5,10
         B    CTEOR
         DORG*-4
XNUL     TFM
SPP1     DS   ,*
         B    *-32
         DORG*-4
         AM   SPO,3,10
         B    CTEOR
         DORG*-4
         DS   2
MAR5     TFM  SPV,13,610
         B    DEMUL-1,,6
         DORG*-4
         DS   4
ERMUL    BT   BUTOIR,*-1
         TDM  PIE
         TDM
         B    DEMUL-1,,6
         DORG*-4
         DS   4
ERMUL1   BT   BUTOIR,*-1
         TDM  PIE
         B    DEMUL-1,,6
         DORG*-4
         DS   5
FORMATTF  ESEIN,*-1
         TF   SCT,*-13
         AM   SCT,5,10
         BTM  FLSV
         BNE  *-40,SCS,611
         TF   ERMUL&30,SCS
         BTM  DEMUL,*&12
         BT   FLSV,ESEIN
         BTM  BUTOIR,6215,811
         B    *-100,,6
         DORG*-4
FORSTABNE  *&44,PIE
         TDM  PIE,0
         BTM  BUTOIR,6211,811
         B    PARA2
         DORG*-4
         TF   ERMUL&30,SCS
         BTM  DEMUL,*&12
         TFM  SE,PIE
         B    APEL1
         DORG*-4
```

```
****    ZONES DE TRAVIL DES PROCE D E/S

13   DC   13,1000000000000,
     DC   13,5000000000000,
12   DS   2
RZ   DS   2
RNZ  DS   ,DERZ
G    DS   2
     DS   2
B    DS   2
Z    DS   2
     DS   2
     DS   2
     DS   2
     DS   2
     DS   2
     DS   2
     DS   2
     DC   2,0,
     DS   1
E    DS   1
T    DS   1
M1   DS   1
P1   DS   1
PT   DS   1
M2   DS   1
P2   DS   1
     DSC  1,0
11   DS   ,F11
R    DAS  17
Y    DAS  81
D    DAS  81
     DAC  4,VRAI,
AI   DS   ,*
     DAC  4,FAUX,
UX   DS   ,*
     DC   10,5000000000,
10   DS   2
     DS   1
     DS   1
     DS   1
     DS   1
     DS   1
RA   DS   6
     DS   1
     DC   5,0
AC   DS   ,*-4
     DC   6,0,
     DC   1,@
N    DAS  81
```

```
BLANCSDAC 1, ,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
```

```
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DC   2,0,
     DAC  1,@,
```

```
PIF     DAC 1, ,
        DAS 400
MAXV    DAC 1,@,
        DC  2,@,
        DC  2,@,
        DS  200
PO      DS  1
        DS  5
MARO    DSC 2,@,
FORTABDAC 1,B,
        DSA ZONE1 ,ESPACE,ESPACE,ESPACE,CAR5  ,ZONE6
        DAC 1,Z,
        DSA ZONE1 ,NOMBRE,ZER3  ,ERSO7 ,CAR5  ,ZONE6
        DAC 1,D,
        DSA ZONE1 ,NOMBRE,DIG3  ,ERSO7 ,CAR5  ,ZONE6
        DC  2,34,
        DSA ARA1  ,ARABES,ARABES,ARABES,ARA5  ,ARA6
        DAC 1,,,
        DSA VIRG1 ,APEL1 ,APEL1 ,APEL1 ,CAR5  ,VIR6
        DAC 1,/,
        DSA VIRG1 ,LIGNE ,APEL4 ,LIGNE ,CAR5  ,FRTA6
        DAC 1,N,
        DSA ZONEN ,NORMA ,ERSO7 ,ERSO7 ,CAR5  ,ZONN6
        DAC 1,F, '
        DSA ZONE1 ,BOOL  ,ERSO7 ,ERSO7 ,CAR5  ,BOOL6
        DAC 1,L,
        DSA ZONE1 ,LOGIO ,ERSO7 ,ERSO7 ,CAR5  ,BOOL6
        DAC 1,&,
        DSA ZONE1 ,NOMBRE,PLUS3 ,ERSO7 ,CAR5  ,ZONE6
        DAC 1,-,
        DSA ZONE1 ,NOMBRE,MOINS3,ERSO7 ,CAR5  ,ZONE6
        DAC 1,.,
        DSA ZONE1 ,NOMBRE,PT3   ,ERSO7 ,CAR5  ,ZONE6
        DAC 1,E,
        DSA ZONE1 ,NOMBRE,EXP3  ,ERSO7 ,CAR5  ,ZONE6
        DAC 1,T,
        DSA ZONE1 ,NOMBRE,DIG3  ,ERSO7 ,CAR5  ,ZONE6
        DAC 1,F
        DSA VIRG1 ,MAR1  ,MAR1  ,MAR1  ,MAR5  ,MAR6
        DAC 1, ,
        DSA RAF   ,RAF   ,RAF   ,RAF   ,RAF   ,RAF
        DAC 1,%,
        DSA RAF   ,RAF   ,RAF   ,RAF   ,PARO  ,RAF
        DAC 1,п,
        DSA RAF   ,RAF   ,RAF   ,RAF   ,PARF  ,RAF
        DAC 1,X,
        DSA RAF   ,RAF   ,RAF   ,RAF   ,FORX  ,RAF
        DAC 1,S,
        DSA VIRG1 ,SAUTY ,APEL4 ,SAUTY ,CAR5  ,FRTA6
        DC  2,@,
```

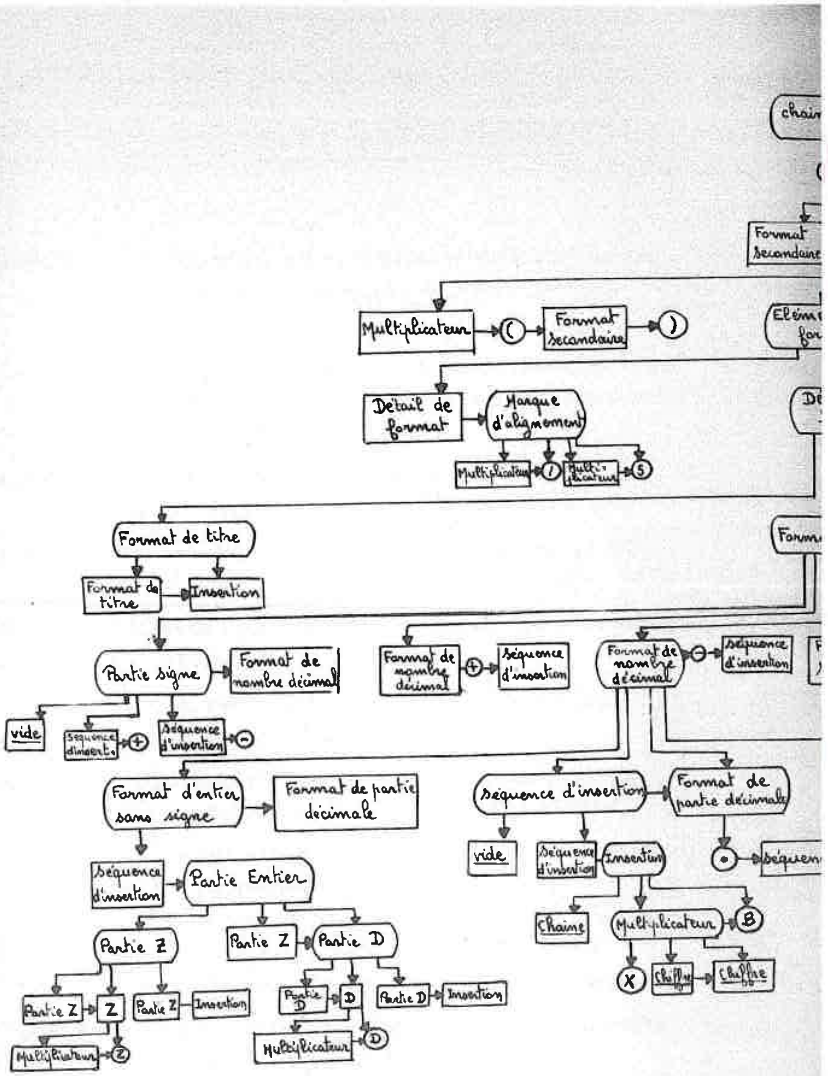
```
****    PROCEDURES USUELLES TRANSLATABLES

****    FONCTION SIN ET COS

        DORG10000
COS     TEM ADES,*-1,17
        R   INDES
        DORG*-4
        TDM *&127,0,0
        B   *&44,,0
        DORG10000
SIN     TEM ADES,*-1,17
        R   INDES
        DORG*-4
        TDM CLE&11,1,011
        TEL EAC,OUT,11
        CE  EAC-11
        TDM EAC-12,0,11
        TDM CLE&10,1,0
        BNE *&48,EAC-2,0
F       CE  EAC-2,,0
        BNE *&24,CLE&11,01
        TDM CLE&10,0,0
        TE  IMSA&10,9SPE&1
        CM  EAC,0,10
        BNL POZEXP,,0
        CM  EAC,10,1011
        BL  BRD,,0
        TEM *&35,EAC+1,07
        A   *&23,EAC,0
        A   IMSA&10
ID      BD  *&24,CLE&11,01
        A   IMSA&10,PIOV2
        S   IMSA&10,TWOPI,1
        BH  *-12,,0
        SE  IMSA-1
        A   IMSA&10,PI,1
        BN  *&36,,0
        SM  CLE&10,01,010
        SE  IMSA&10
        A   IMSA&10,PIOV2
        TE  79,ZERO-74
        M   IMSA&10,IMSA&10
        TE  IMSA&10,88
        SE  IMSA-1
        TE  EAC,ONEZ&2
        TR  AAB-3,AABS+8,01
ARC     TDM SUM&1,2,0
        TE  99,88
        TEM 87,0000,8
        D   88,ABS,1
        BZ  DONE,,0
        TE  SAVE,95,10
JM      S   EAC,SAVE
        AM  AAB-2,02,010
        AM  AAB,02,010
        M   AAB,AAB-2,01
        TE  ABS,99,0
        TE  79,ZERO-74
        M   IMSA&10,SAVE
        CM  SUM&1,9521,08
        BE  OVER,,0
        TDM SUM&1,1,0
        B   OVER&12,,0
        DORG*-4
DONE    C   EAC-1,9SPE
        BE  ZEREAC
        TEM SAVE,01,10
        BD  NORM&72,CLE&10,1
        SE  99
        B   NORM&72
        DORG*-4
POZEXP  CM  EAC,10,10
        BH  *&56,,0
        TEM *&30,IMSA&9,07
        S   *&18,EAC,0
        TE  ,EAC-2
        B   BRD,,0
        DORG*-4
        TEM EI,670,0
        B   UNFLO
        DORG*-4
AAB     DS  4
ABS     DS  4
        DS  1
        DC  2,01
        DC  2,02
AABS    DC  5,0002@,
TWOPI   DC  12,628318530718,
PI      DC  12,314159265359,
        DC  5,0,
*
******  FONCTION ARCTAN
*
        DORG10000
FATN    TEM ADES,*-1,17
        R   INDES
        DORG*-4
        TEL EAC,OUT,11
        BD  *&20,EAC-11,0
        B   ZEREAC
        DORG*-4
        TEM TEST&11,,08
        BNE TEST&24,EAC-2,0
TEST    SE  *&9,,0
        CE  EAC-2
        CM  EAC,0,10
        BH  TEST1&20,,0
        BE  ALPH&12,,0
        CM  EAC,45,10
        BL  FOO,,0
        TEM ALPH-1,EAC-2,07
        A   ALPH-1,EAC,0
        TE  EAC-12,9SPE-1
        CE  EAC-11
        TE  EAC-2
ALPH    SE  EAC-11
        CM  EAC-10,29,10
        BL  *&132,,0
        TE  97,9SPE-1
        MM  EAC-2,06,10
        TDM 88,1,11
        S   EAC-2,SIX,1
        TE  BETA,99
        TE  79,ZERO-42
        LD  88,EAC-2
        D   99,BETA
        TE  EAC-2,87
        SE  TEST&11,,0
        TE  79,ZERO-44
        M   EAC-2,EAC-2
        TE  SAVE,89
        TE  BETA,9SPE
        TEM ATN1&35,10,010
        TEM ALPH&11,9,010
ATN1    TE  99,ZERO-47
        TEM 88,10,10
        DM  88
        S   96,BETA
        TE  BETA,96
        TE  79,ZERO-43
        M   BETA,SAVE
        TE  BETA,89
        SM  ATN1&35,2,010
        SM  ALPH&11,01,010
        BNZ ATN1,,0
        TE  SAVE,ONEZ&2
        S   SAVE,BETA
        TE  79,ZERO-42
        M   SAVE,EAC-2
        BNE TEST2,TEST&11,01
        BNN TEST2-12,,0
        SE  89
        A   89,TAN6,1
TEST2   BNE *&36,TEST&10,01
        SE  89
        A   89,PIOV2
        TEM SAVE,01,10
        TE  EAC,81
        BD  TEST1-24,EAC-12,0
        TR  EAC-12,EAC-11
        TDM EAC,0
        SM  SAVE,01,10
        B   *-48,,0
        DORG*-4
        SE  EAC-12
        TE  EAC-2,EAC-3
TEST1   TD  99,TEST&9,1
        B   ENDD.
        DORG*-4.
        SE  TEST&10,,0
        TE  99,ZERO-22
        TE  89,ONEZ
        D   89,EAC-2
        TE  EAC-2,89
        BD  *&44,EAC-12,0
        SE  EAC-11
        SM  EAC,01,10
        B   *&32,,0
        DORG*-4
        TE  EAC-2,EAC-3
        SM  EAC,02,10
        BL  *&44,,0
        BE  ALPH&12,,0
        SE  EAC
        B   TEST&60,,0
        DORG*-4
        TEM SAVE,00,10
        TE  EAC-2,PIOV4,1
        B   TEST1,,0
        DORG*-4
        TD  99,TEST&9,1
        B   ENDD&12
        DORG*-4
FOO     BNE *-20,TEST&10,01
        TE  IMSA&10,EAC-2
        TE  EAC-2,PIOV2-1
        CM  EAC,10,1011
        BNH *&72,,0
        CE  IMSA&1
        TDM IMSA,,11
        TEM *&35,IMSA&10,07
        A   *&23,EAC,0
        S   EAC-2
        TEM SAVE,01,10
        B   TEST1-12,,0
        DORG*-4
SIX     DC  10,6000000000,
PIOV4   DC  10,7853981634,
TAN6    DC  11,54041950027,
        DC  6,0,
*
******  FONCTION EXP
*
        DORG10000
FEXP    TEM ADES,*-1,17
        R   INDES
        DORG*-4
        TEL EAC,OUT,11
        TEL BETA,ONEZ&6
        BD  *&20,EAC-11,0
        B   GOBACK-24,,0
        DORG*-4
```

```
        TD   MU-1,FAC-2
        CE   FAC-2
        TE   79,ZERO-42
        M    FAC-2,LOGE,1
        TE   SAVE&2,FAC
        TD   FAC&4,FAC&1
        TE   FAC&3,82
        BD   *&48,FAC-11,0
        TR   FAC-11,FAC-10
        SM   SAVE&2,01,10
        SE   FAC-11
        CM   SAVE&2,10,1011
        BNH  GOBACK-24,,0
        BV   *&12,,0
        AM   SAVE&2,0,10
        BE   CALC,,0
        BL   CALC&68,,0
        CM   SAVE&2,02,10
        BH   E12,,0
        BL   *&44,,0
        A    BETA,FAC-10
        TR   FAC-12,FAC-10
        B    CALC-24,,0
        DORG*-4
        TDM  FAC-12,0,11
        CE   FAC-11
        A    BETA,FAC-11
        TR   FAC-11,FAC-10
        BV   E12,,0
        SE   FAC-11,,10
LC      TEM  *-1,0,010
        CM   FAC-10,34,10
        BL   GORD,,0
        SM   FAC-10,34,10
        AM   CALC-1,01,010
        B    CALC&12,,0
        DORG*-4
        TEM  *&59,FAC,07
        A    *&47,SAVE&2,0
        TE   FAC-12,9SPE-1
        CE   FAC-11
        TE   FAC
        SE   FAC-11
)RD     TE   79,ZERO-20
        M    FAC,LOG10-1
        SE   77
        TE   SAVE,89
        TE   FAC&2,80
        TEM  FACT&11,02,010
        A    BETA-2,SAVE
        TE   79,ZERO-38
        M    SAVE,FAC&2
        TE   99,86
        TEM  86,0,10
CT      DM   87
        BZ   FACT&56,,0
        TE   SAVE,87
        AM   FACT&11,01,010
        B    FACT-60,,0
        DORG*-4
        CM   CALC-1,0,010
        BE   *&80,,0
        TE   79,ZERO-38
        M    BETA-4,TEN34,1
        SE   77
        TE   BETA-4,88
        SM   CALC-1,01,010
        B    FACT&68,,0
        DORG*-4
        BNE  GOBACK-24,MU-1,0
        TE   99,ZERO-21
        TE   89,ONE&2
        D    90,BETA-4
        BV   *&44,,0
        TE   BETA-6,87
        SM   BETA,01,10
        B    *&20,,0
        DORG*-4
        SM   BETA,02,10
        BE   *&24,,0
        SE   BETA
        TE   FAC-2,BETA-6
        TE   FAC,BETA
GOBACKTD     FAC&1,BETA&1
        B    ERXVE30,,6
        DORG*-4
E12     TD   FAC&1,BETA&1
        TEM  E1,674,0
        BNE  OVELO,MU-1
        B    UNELO
        DORG*-4
TEN34   DC   12,218776162295,
LOGE    DC   12,434294481903,
        DC   5,0,
*
******     FONCTION LOG
*
        DORG10000
FLN     TEM  ADES,*-1,17
        B    INDES
        DORG*-4
        TEL  FAC,OUT,11
        BD   *&44,FAC-11,0
        TEM  E1,672,0
        SE   99
        B    OVELOW
        DORG*-4
        TEM  GOAL&18,ENDD&12,07
        BNE  *&48,FAC-2,0
        TEM  E1,673,0
        TEM  GOAL&18,ERROR,07
        CE   FAC-2
        TEM  CORN-25,9SPE&2,07
        TE   CORN-13,FAC,0
        CE   FAC-11
        TDM  FAC-12,1,11
        CM   FAC-11,15,10
        BNL  *&44,,0
        A    FAC-2,FAC-2
        AM   CORN-25,13,010
        B    *-60,,0
        DORG*-4
        TE   BETA,TW07,1
        S    BETA,FAC-2
        TE   99,ZERO-20
        TE   97,BETA
        D    88,FAC-2
        TE   FAC,88
        TE   79,ZERO-39
        M    FAC,FAC
        TE   SAVE,87
        A    FAC,FAC
COUN    TEM  IMSA-4,0,10
        TEM  COUN&71,10,010
        TE   BETA,9SPE&1
        TE   99,ZERO-10
        TEM  88,10,10
        DM   88
        A    BETA,97
        TE   79,ZERO-39
        M    BETA,SAVE
        TE   BETA,87
        SM   COUN&71,02,010
        SM   IMSA-4,01,10
        BNZ  COUN&36,,0
        TE   79,ZERO-39
        M    BETA,FAC
        A    97,FAC
        TE   FAC
        SE   FAC
        S    FAC,87
CORN    TE   92,9SPE-1
        M    LOG10,CORN-13,1
        A    99,FAC
        AM   97,0,10
        BZ   ZERFAC
        TEM  SAVE,03,10
        TE   FAC&3,99
        CE   FAC&3
        TD   FAC&4,BETA&1
        BD   *&48,FAC-11,0
        TR   FAC-11,FAC-10
        TDM  FAC&3,0
        SE   FAC-11
        SM   SAVE,01,10
        B    *-60,,0
        DORG*-4
        TD   FAC&1,BETA&1
GOAL    TE   FAC,SAVE
        B    ENDD&12
        DORG*-4
TW07    DC   11,20000000000,
        DC   6,0,
*
******     FONCTION RAC2
*
        DORG10000
SQRT    TEM  ADES,*-1,17
        B    INDES
        DORG*-4
        TM   OUT,50,&10
        TEL  FAC,OUT,11
        BD   SQ1,FAC-11,0
        B    ZERFAC
        DORG*-4
SQ1     TDM  FAC-12,0,11
        TDM  SQFX&13,2,0
        BNE  *&48,FAC-2,0
        CE   FAC-2
        TDM  SQFX&13,0,0
        TEM  E1,676,0
        BD   SQ3,88,0
        TEM  SQ2&42,87,07
        BNE  SQ2&12,00,0
SQ2     SE   97
        TE   FAC,97
        LD   80,9SPE
        TE   ,FAC-2
        TEM  LOOP&18,77,07
        TEM  LOOP&23,FAC-12,07
        TEM  LOOP&102,77,07
        TE   FAC-2,ONE&61
        B    LOOP&108,,0
        DORG*-4
LOOP    AM   LOOP&23,2,0&10
        S
        BNM  LOOP,,0
        CM   LOOP&23,FAC+2,07
        BNL  SQFX-48,,0
        A    LOOP&18,LOOP&23,01&11
        CE   LOOP&18,,06
        AM   *&18,1,010
        SE
        AM   LOOP&18,02,010
        AM   LOOP&23,01,010
        SM   LOOP&23,9,0&10
        B    LOOP&12,,0
        DORG*-4
        LD   80,9SPE
        MM   FAC-2,50,10
```

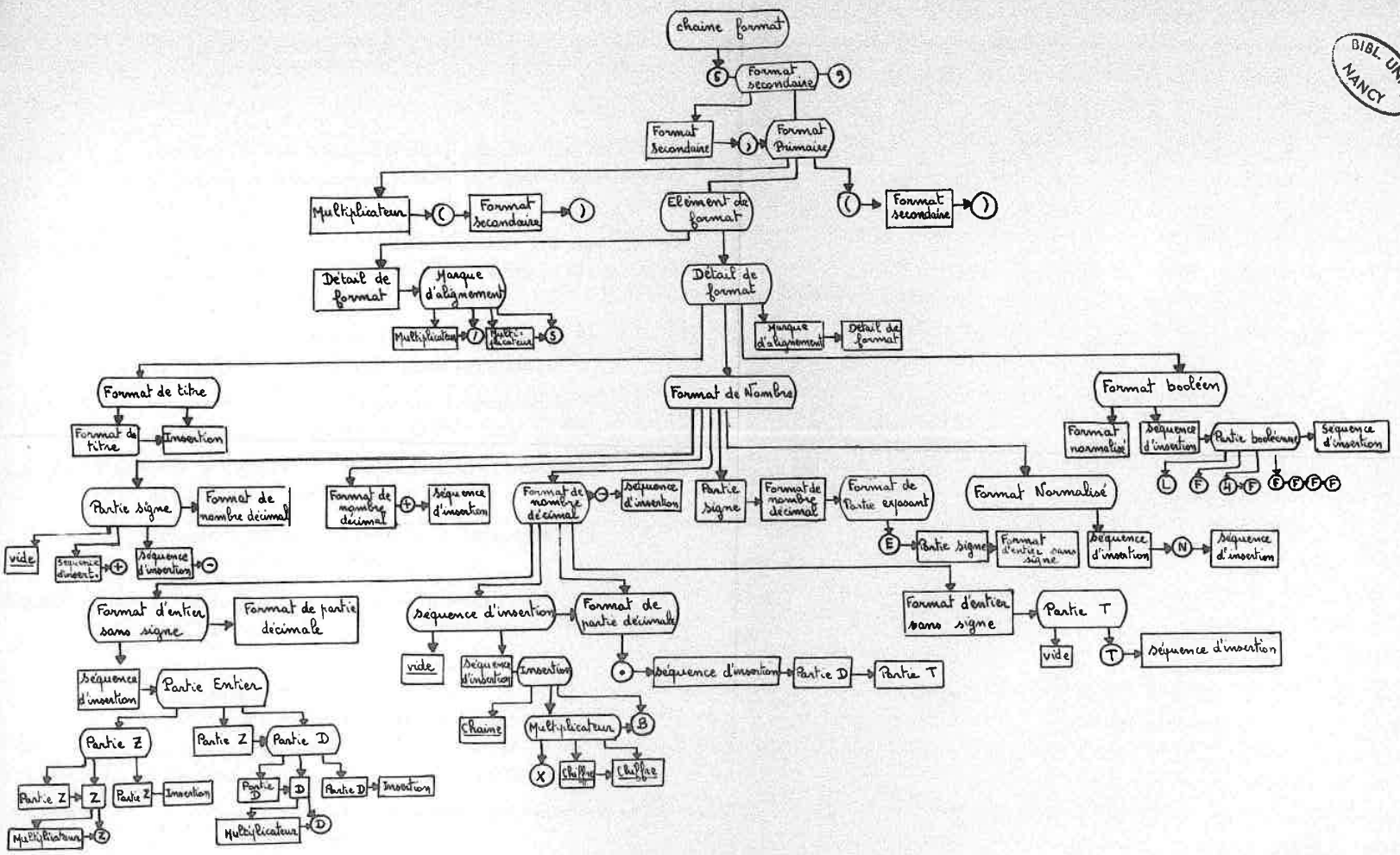
```
       TF   FAC-2,97
       SF   FAC-11
SOEX   TFL  OUT,FAC,6
       BR   ERROR
       DORG*-4
SO2    AV   99,50,10
       CF   FAC-11
       TFM  SO2&42,88,07
       BNN  SO2&12,,0
       B    SO2,,0
*
******     FONCTION ENTIER
*
       DORG10000
ENTIERTFM  ADES,*-1,17
       B    INDES
       DORG*-4
       TFL  CC,OUT,11
       CF   ELBR
       TFM  SOBB&32,*&20,17
       B    BF
       DORG*-4
       TF   OUT,TR,6
       BR
*
******     FONCTION SIGNE
*
       DORG10000
SIGNE  TFM  ADES,*-1,17
       B    INDES
       DORG*-4
       TF   CC,OUT,11
       BNF  SIGNEN,CC-1,0
       TFL  CC,OUT,11
       BD   *&26,CC-11,0
       TF   OUT,ZEROEX,6
       BR
       DORG*-10
       BNF  *&26,CC-2,0
       TF   OUT,UNEGEX,6
       BR
       DORG*-10
       TF   OUT,UNEX,6
       BR
       DORG*-10
SIGNENCM   CC,0,10
       BF   SIGNE&68,,0
       BH   SIGNE&108,,0
       B    SIGNE&88,,0
*
******     FONCTION ABS
*
       DORG10000
ABSO   TFM  ADES,*-1,17
       B    INDES
       DORG*-4
       TF   CC,OUT,11
       BNF  ABSEN,CC-1,0
       TFL  CC,OUT,11
       CF   CC-2
       TFL  OUT,CC,6
       BR
       DORG*-10
ABSEN  CF   OUT,,6
       BR
*
******     FONCTION CLEF
*
       DORG10000
CLEF   TFM  ADES,*-1,17
       B    INDES
       DORG*-4
       TF   CC,OUT,11
       BNF  CLENT,CC-1,0
       CM   CC,1,10
       BNF  CLBAS,,0
       TFL  CC,OUT,11
       BNF  *&26,CC-2,0
CLBAS  TDM  OUT,1,6
       BR
       DORG*-10
       SF   CC-10
       CM   CC-2,0,10
       BNF  CLBAS,,0
       CM   CC-10,40,1011
       BL   CLBAS,,0
       CF   CC-11
       TD   CLTEST&9,CC-11,0
       B    CLTEST,,0
       DORG*-4
CLENT  CM   CC,0,10
       BNH  CLBAS,,0
       CM   CC,4,10
       BH   CLBAS,,0
       TD   *&21,CC,0
CLTESTBNC1 CLBAS,,0
       TDM  OUT,0,6
       BR
*
******     FIN DES SOUS/PROGRAMMES
*
*
       DEND20000
```



CARTE SYNTAXIQUE

des CHAINES-FORMATS

N.B : Les symboles Algol sont soulignés



CARTE SYNTAXIQUE
des CHAINES - FORMATS

N.B : Les symboles Algol sont soulignés

REFERENCES

1 M.CUSEY — Construction d'un compilateur ALGOL pour IBM 1620, Thèse de 3ème cycle de Mathématiques, Centre de Calcul Automatique de Nancy (1964)

2 J.M.LAPORTE — Traitement des tableaux en ALGOL, extension à l'Algol matriciel, Thèse de 3ème cycle de Mathématiques, Centre de Calcul Automatique de Nancy (1965)

3 C.PAIR — Arbres, piles et compilation, Revue Française de Traitement de l'Information, 7, (1964), pages 199 à 216

4 P.NAUR (Editor) — Revised report on the algorithmic language ALGOL 60. Communications of the ACM, 6, (1963), pages 1 à 17

5 L.BOLLIET, N.GASTINEL et P.J.LAURENT — Un nouveau langage scientifique : ALGOL Hermann, PARIS, (1964)

6 — Traitement des procédures en ALGOL. Communication ultérieure du Centre de Calcul de Nancy.

7 M.GROSS et A.LENTIN — Notions sur les grammaires formelles, Publication n° NMC/18.2.4/AI de l'Institut Blaise Pascal, PARIS (1964)

8 — Bulletin de l'AFCALTI, Revue Française de Traitement de l'Information, 7, (1964), page VIII

9 — - Fortran II - Procédures opérations - Documentation ordinateurs - n° 1620-25/20 1012 PARIS (1963)
- IBM 1620 - Program Library - notice 1620-LM-024

10 C.HASTINGS — Approximations for Digital Computers - Princenton University Press - Princenton 1955

11 J.LEGRAS — Une extension d'Algol : Algol linguistique - Communication faite au Congrès de l'AFCALTI (1964)

12 D.E.KNUTH — A proposal for Input-Output Conventions in ALGOL 60 - A report of the subcomittee on ALGOL of the ACM Programming Languages Committee - Communication of ACM, 7, (1964) pages 273 à 283

13 Report on input-output - Procedures for Algol 60
ACM 7 (64) pages 628 à 630.

14 IBM Unité composantes du 1620 - Brochure
n° 10-139 - Paris - mai 62
IBM Instructions 1620 - Brochure
n° 10-1340 - Paris - janvier 63
IBM Système de programmation symbolique 1620 (SPS)
brochure n° 10-1341 - janvier 63

15 BENSON PERRY & MORTIMER L.MENDELSOHN Picture generation with a standard line printer - Medical applications - Communication of the ACM, 7 (1964) pages 311 à 313.

Vu et Approuvé
Nancy, le
Le Doyen de la Faculté
des Sciences de NANCY

M. AUBRY

Vu et Permis d'imprimer
Nancy, le
le Recteur :
Président du Conseil de
l'Université
P. IMBS