UNIVERSITE DE NANCY — FACULTE DES SCIENCES

TRAITEMENT DES TABLEAUX ALGOL

EXTENSION A L'ALGOL MATRICIEL

THESE

pour l'obtention du

DOCTORAT de SPECIALITE MATHEMATIQUES (3ème CYCLE)

Soutenue devant le Jury le 8 février 1965

par

Jean Marie LAPORTE

JURY : Mr J. LEGRAS Président
Mr C. PAIR Examinateurs
Mr M. DEPAIX

UNIVERSITE DE NANCY FACULTE DES SCIENCES

TRAITEMENT DES TABLEAUX ALGOL

EXTENSION A L'ALGOL MATRICIEL

par

Jean Marie LAPORTE

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.			
RION	Chimie biologique	SCHWARTZ	Exploit. minière
ELSARTE	Analyse supérieure	GAYET	Physiologie
OUBAULT	Géologie	MANGENOT	Phytopathologie
EILLET	Biologie animale	MALAPRADE	Chimie
CHEVIN	Botanique	HADNI	Physique
ARRIOL	Chimie théorique	BONVALET	Mécanique physique
IZETTE	Physique	KERN	Minéralogie
UILLIEN	Electronique	BASTICK	Chimie
IBERT	Chimie physique	DUCHAUFOUR	Pédologie
EGRAS	Mécanique rationnelle	NEEL	Chimie ind. orga.
OLFA	Minéralogie	GARNIER	Agronomie
ICLAUSE	Chimie	WEPPE	Minéralogie appli.
AIVRE	Physique appliquée	BERNARD	Géologie appliquée
UBRY	Chimie minérale	CHAMPIER	Physique
UVAL	Chimie	REGNIER	Physico-chimie
OPPENS	Radiogéologie	GAY	Chimie biologique
RUHLING	Physique	WERNER	Botanique
UHNER	Physique expérimentale	CONDE	Zoologie
ILLY	Géologie	STEPHAN	Zoologie
E GOFF	Génie chimique	EYMARD	Cal. Dif. et Int.
HAPON	Chimie biologique	LEVISALLES	Chimie organique
EROLD	Chimie industrielle	N.	Physique
		N.	M. M. P.

MAITRES DE CONFERENCES ET PROFESSEURS SANS CHAIRE

MM.			
OSSE	Génie chimique	Mme HERVE	Math. (propédeutique)
OCCI	Géologie	AUROUZE	Géologie
UILLAUME	Psychophysiologie	MARI	Chimie I. S. I. N.
LAN	Mécanique physique	LAFON	Physique I. S. I. N.
me BASTICK	Chimie M. P. C. Epinal	FELDEN	Phy. Théor. & Nucl.
UDEFIN	Physique	FLECHON	M. P.C.
ORN	Physique	VIGNES	Physique (Mines)
RENTZ	Biologie animale	Melle HUET	Math. (S. P. C. N.)
		JANOT	M. P. C. Epinal

SECRETAIRE PRINCIPAL : C. CARON

Je remercie Monsieur le Professeur J. LEGRAS, Directeur du Centre de Calcul, pour l'orientation qu'il a donné à mes deux années d'études au Centre de Calcul de Nancy. Ainsi que Monsieur C. PAIR, pour les conseils et l'attention qu'il m'a apporté, dans la conception et la rédaction de mon travail.

Je remercie Monsieur M. DEPAIX de me faire l'honneur de participer à mon jury, et Monsieur PROCYK, Ingénieur I. B. M. qui m'a permis de travailler dans de bonnes conditions sur l'ordinateur I. B. M. 1620.

SOMMAIRE

CHAPITRE I

GENERALITES SUR LE COMPILATEUR

1. Le compilateur ALGØL étudié, a été construit pour un ordinateur I. B. M. 1620 à 60 000 positions de Mémoires. Ayant tous les dispositifs spéciaux (Adressage indirect, virgule flottante automatique). Il a été programmé en S. P. S. (Système de programmation symbolique 1620, Janvier 1963. 10. 1341), qui permet la codification, dans un langage symbolique plus significatif et plus facile à traiter que le langage machine numérique.

2. Le compilateur a été construit en équipe, pour la plus grosse partie par M. CUSEY [5] ; toutes les notations utilisées dans la programmation des tableaux locaux et rémanents, ont été définies dans sa thèse ; les plus importantes seront rappelées. J. ANDRE [4] a traité les nombres, les fonctions standard et les entrées-sorties.

La partie théorique est l'oeuvre de C. PAIR [3] qui nous a dirigés dans nos travaux.

3. Rappels de Notations

Le programme ALGØL est perforé sur cartes qui sont lues en séquence en mode alphanumérique à partir de (SE), leur contenu est analysé, grâce à un compteur (MEMSE) de 5 positions qui contient l'adresse "moins un" du prochain caractère à identifier. Un sous-programme (DECAL) rend dans (MEMSE) l'adresse de la zone réduite au caractère suivant.

PILES :

Pile E : La suite d'entrée donnée par le programme ALGØL sera appelée "pile E" : (MEMSE) indique à chaque instant l'adresse du sommet de la pile.

Pile O : Elle reçoit les équivalents numériques des symboles de base. A chaque instant, l'adresse de la zone située au sommet de cette pile est dans le compteur (MEMSO). La pile O se remplit suivant les adresses croissantes. Le sommet de cette pile est désigné dans la suite par S(O).

Pile V : Elle reçoit les adresses des zones affectées aux identificateurs, nombres, variables, ainsi que leur type I.

Un élément de V s'écrira αI avec :

- α: adresse qui occupe 5 positions.
- I : le type pour une position.

I prend les valeurs :
- 0 pour un type réel
- 1 pour un type entier
- 2 pour un aiguillage
- 3 pour un type booléen
- 4 pour une étiquette.

Pour les tableaux :
- $\overline{5}$ Tableau de type réel
- 5 Tableau de type entier
- $\overline{6}$ Tableau de type booléen

6 sera utilisé en ALGOL Matriciel, pour indiquer qu'un tableau à 2 dimensions, donc une matrice, est un résultat intermédiaire. D'autres valeurs seront utilisées pour les Procédures [6].

Lorsque I = 0, 1, 3, 4 on met un "Flag" lorsque l'adresse α indique un résultat intermédiaire.

Les opérations sur la pile V se font à l'aide d'un compteur (MEMSV) qui indique, en dehors des Σ et Σ' (séquences de programmes ou sont construits les Γ et $\Gamma'[5]$) l'adresse du type de l'élément situé au sommet S (V) de la pile V. La pile se remplit suivant les adresses décroissantes.

La pile V est aussi utilisée pour stocker :

- En début de Bloc les contenus du compteur d'affectation (MRINT) et de l'indicateur d'adresse de table (C P T R 2). Les adresses ainsi stockées seront renvoyées dans ces compteurs à la fin de la compilation du bloc. Les zones réservées aux variables locales et les positions utilisées de la table seront aussi libérées.

- Lors d'une déclaration l'adresse dans la table où on doit stocker le type I de l'identificateur déclaré.

TABLE DES IDENTIFICATEURS

Les identificateurs du programme ALGØL sont rangés dans une table, ainsi que leurs "adresses" (par exemple pour un réel ou un entier, l'adresse de la zone qui lui sera affectée à l'éxecution, pour un tableau l'adresse d'un enregistrement ou seront stocké des renseignements), et leur type.

Une "ligne" de la table occupe 22 positions, utilisées de la manière suivante:

- l'identificateur occupe 14 positions (7 caractères alphanumériques en sont conservés).

- l'adresse correspondante a occupe 5 positions,

- le type I occupe 1 position.

- Une zone de deux positions est utilisée par les étiquettes et les aiguillages pour garder le "niveau de procédure" ; dans la version du compilateur n'acceptant que les tableaux locaux, pour un identificateur de tableau, cette zone indiquera la dimension du tableau c'est-à-dire le nombre de ses paires de bornes ; dans la version qui accepte les tableaux Rémanents : 1 position indiquera si le tableau est rémanent ou non :

0 : tableau local

9 : tableau Rémanent

l'autre position la dimension du tableau.

Un compteur (CPTR2) indique l'adresse de la dernière position de la dernière ligne construite dans la table.

FORMATION de la TABLE

Si le sommet S (O) de la pile O est un déclarateur autre que aiguillage ou procédure (qui sont traités différemment) :

- On introduit l'identificateur dans une nouvelle ligne de la table.

- On envoie sur la pile V l'adresse de la table à laquelle on doit ranger le type de ces identificateurs. Le type et l'adresse (à l'exécution) affectés à l'identificateur dans le Γ'(S (O) ;) ou le Γ'(S (O),) suivant. Cette adresse "d'exécution" sera prise dans un compteur d'affectation de zone, (MRINT) s'il s'agit d'une variable locale, ou d'un tableau local, (REMAN) s'il s'agit d'une variable rémanente ou d'un tableau rémanent (reconnaissable par le symbole situé dans S (O) qui sera alors REMANENT) le compteur utilisé regressera ensuite d'une quantité en liaison avec le type. Ainsi pour un Entier ou Réel de 12 positions, un booléen 1 position et un tableau local 10 (N +1) positions, si N est la dimension du tableau.

UTILISATION de la TABLE

Si S (O) n'est pas un déclarateur, l'identificateur se trouve stocké dans la table. On le recherche alors par un sous-programme de consultation de table (CØNSUL) dont le but est de :

- envoyer en S (V) l'adresse correspondante à l'identificateur, et le type

- laisser dans un compteur (CPTR3) jusqu'a la prochaine consultation de table, l'adresse de la dernière zone de la "ligne". Ainsi pour un identificateur de tableau CPTR3 indiquera la dimension du tableau et sera mis sur S (V) au moment de la compilation d'une variable indicée, pour comparer la dimension du tableau déclaré au nombre d'indices de cette variable indicée. Dans le cas des procédures dont un identificateur de tableau est paramétre formel, la première fois qu'il sera rencontré, le nombre d'indice sera mis en (CPTC3) au bout de la ligne, et les fois suivantes le nombre d'indices recalculé devra être comparé à ce nombre. S'il n'y a pas égalité un message d'erreur sera imprimé.

PORTEE DE LA TABLE

Un identificateur ne pouvant être utilisé à l'extrémité du bloc dans lequel il a été déclaré, il ne restera pas dans la table après la compilation de ce bloc.

De même les zones d'exécution réservées pour les variables locales seront réinitialisées de la façon suivante :

Quant on entre dans un bloc, c'est-à-dire à l'identification de DEBUT suivi d'un déclarateur, on garde sur la pile V les contenus de (CPTR2) et de (MRINT). A la fin de la compilation du bloc, on réinitialisera (CPTR2) et (MRINT) avec les adresse envoyées précédemment dans V.

PROGRAMME OBJET :

Le programme objet est construit par concaténation ([5], chapitre I) de séquences appelées Γ ou Γ': lorsqu'un symbole x entre dans la pile O, on exécute une séquence du programme de compilation $\Sigma'(S(O)x)$ qui dépend du sommet de la pile O, et de x ; quand S (O) sort de la pile O, on éxécute une séquence $\Sigma(S(O))$, qui ne dépend que de la pile S (O).

Ces séquences déterminent une régle de grammaire par différents tests ([5], chapitre III) construisent le Γ' ou le Γ correspondant, et placent certaines adresses sur la pile V. Par exemple, à la compilation d'une variable indicée, on mettra sur V l'adresse où sera rangée à l'exécution l'adresse de cette variable indicée.

Les Γ et Γ' sont construits dans une zone de travail (C4) et stockés en (C3). L'adresse à partir de laquelle on peut stocker une nouvelle séquence est dans le compteur (MPGØB).

Le programme sera perforé carte par carte, et non stocké en entier dans la mémoire ([5] page 47).

TABLEAU

S(E) / O)	OA	[	,	:	]	;	ENTIER	REEL	BOOLEEN
OA		3	1	1	1	2	1	2	2	2
[	0	3	3	3	1	2	2	2	2	2
:	0	3	1	2	1	2	2	2	2	2
AU	2	2	3	2	2	2	1	2	2	2
;	2	3	2	2	2	0	1	0	0	0
TIER	2	2	3	2	2	0	1	2	2	2
REEL	2	2	3	2	2	0	1	2	2	2
LEEN	2	2	3	2	2	0	1	2	2	2

OA : Opérateurs arithmétiques

REPRESENTATION EN MEMOIRE

S(O) \ S(E)	OA	[	,	:	]	TABLEAU	;	ENTIER	REEL	BO
OA		<	>	>	>		>			
[	<	<	<	<	>					
,	<	<	>	<	>	<	>	<	<	
:	<	<	>		>					
TABLEAU			<				>			
;		<				<		<	<	
ENTIER			<			<	>			
REEL			<			<	>			
BOOLEEN			<			<	>			

OA : OPERATEURS ARITHMETIQUES

MATRICE DE PRIORITE PARTIELLE

CHAPITRE II

TABLEAUX NON REMANENTS

Dans ce chapitre on étudiera la compilation des tableaux par un compilateur n'acceptant pas les tableaux rémanents.

1. DEFINITIONS (d'après [7], [9]) :

Une déclaration de tableau sert à préciser l'identificateur du tableau, la dimension du tableau (c'est à dire le nombre d'indices), le domaine parcouru par chaque indice, et le type des composantes du tableau, réel, entier, ou booléen (le type réel peut être omis).

Exemple : REEL TABLEAU T [$a_1 : b_1$, $a_2 : b_2$] ;

On apprend par cette déclaration que le tableau T est de dimension 2, le premier indice peut aller de a_1 à b_1, le deuxième de a_2 à b_2, et les composantes de ce tableau sont des nombres réels.

$a_1 : b_1$ est appelé une paire de bornes. Lorsque sera utilisée la variable indicée T [I,J] , il faudra que l'on ait $a_1 \leq I \leq b_1$, $a_2 \leq J \leq b_2$, pour que les composantes du tableau soient définies.

Plusieurs tableaux qui ont même dimension, même paires de bornes pour chaque indice et des composantes de même type peuvent être déclarés en une seule fois. Cette partie de déclaration s'appelle une section de tableaux .

Exemple : T1, T2, T3 [1 : 5, 1 : 10];

indique que T1, T2, T3, sont des identificateurs de tableaux à composantes réelles, de dimension 2, le premier indice allant de 1 à 5 et le deuxième de 1 à 10.

Si l'on veut déclarer plusieurs tableaux de même type, on est libre de répeter le symbole de base TABLEAU, ou non. Par contre si les types sont différents, il faut répeter TABLEAU précédé de la déclaration de type convenable.

Exemple : TABLEAU T1, T2 [1:5, 1:10] T3, T4, T5 [-2:0, 0:20] ;

Enfin un tableau déclaré n'est défini que si toutes les bornes inférieures sont plus inférieures où égales aux bornes supérieures correspondantes.

Dans une déclaration de tableau les paires de bornes peuvent être fournies sous forme d'une expression arithmétique : tous les identificateurs intervenant dans ces expressions doivent être déclarés à l'extérieur du bloc ou figure la déclaration : ces expresssions seront calculées chaque fois que l'on entre dans le bloc. C'est à dire avec les valeurs numériques des identificateurs au moment de cette entrée (et non au moment ou on utilise réellement la variable indicée correspondante).

Dans les déclarations de tableaux du bloc le plus extérieur d'un programme toutes les paires de bornes doivent être évidemment des constantes.

2. NOTATIONS :

Les bornes inférieures seront notées $a_1, a_2 \ldots a_n$,

les bornes supéricures : $b_1, b_2 \ldots b_n$,

N sera la dimension du tableau, c'est à dire le nombre de ces paires de bornes.

La longueur de variation du k^e indice sera $c_k = b_k - a_k + 1$.

3. PROBLEMES DE RESERVATION :

3. 1. Généralités

3. 1. 1. Zone réservée au tableau : Les bornes des tableaux pouvant être des expressiosn arithmétiques, la longueur à réserver aux composantes ne pourra être connue à la compilation, il faudra donc construire des ordres qui permettront de calculer cette réservation à l'exécution. Ce sera donc une réservation dynamique.

3. 1. 2. Calcul d'une variable indicée : Certains renseignements sont nécessaires pour calculer l'adresse de chaque composante d'un tableau. Ces renseignements sont donnés à la déclaration, donc à ce moment sera construit un "vecteur de renseignement" [1] qui permettra le calcul d'une variable indicée, connaissant ses indices.

La longueur allouée à ce "vecteur de renseignement" sera connue au moment de la compilation. Donc cette réservation se fera à la compilation comme pour une variable locale. Et le remplissage de cette zone se fera à l'exécution, grâce à des ordres construits à la compilation.

3. 1. 3. A chaque identificateur de tableau, sera donc rattaché un "vecteur de renseignement", il faudra donc garder l'adresse du "vecteur de renseignement". Dans la table, en face de l'identificateur de tableau sera donc mise l'adresse du "vecteur de renseignement" et à chaque appel d'une variable indicée correspondante, l'adresse du "vecteur de renseignement" prise dans la table sera placée sur S (V), de manière à pouvoir créer des ordres contenant cette adresse.

3. 2. Adresse du "vecteur de renseignement":

Cette adresse, et la zone réservée aux renseignements n'est nécessaire que pendant la compilation du bloc étudié ; à sa sortie, l'adresse pourra être perdue et la zone réinitialisée. Donc :

a) La zone allouée au "vecteur de renseignement" sera prise (MRINT.)

b) l'adresse du "vecteur de renseignement " sera mise dans la table avec son type. Et sera perdue de la même façon que pour les variables simples, à la sortie d'un bloc.

3. 3. Structure du "vecteur de renseignement" :

Chaque section de tableau, sera traitée séparément.

a) Cas ou une section de tableau ne contient qu'un seul tableau.

TABLEAU $T\,[a_1 : b_1, a_2 : b_2, \ldots a_n : b_n]$

A partir de l'adresse correspondant à T dans la table, (c'est à dire l'adresse du "vecteur de renseignement") sont rangés :

- adresse minimum des composantes c'est à dire $T\,[a_1, a_2 \ldots a_n]$ sur 5 positions.
- deux positions dont le rôle sera expliqué au b) elles contiendront ici 00
- indicatif de type du tableau : 1 position
- dimension du tableau : 2 positions.

- a_1
- c_1
- a_2
- c_2
- ⋮
- a_n
- c_n

: 5 positions chacun.

Au total $\boxed{10\,(N+1)}$ positions seront occupées par le vecteur de renseignement.

b) Cas général :

Soit une section de Tableaux contenant p identificateurs de tableaux T0, T1, ... Tp-1, le "vecteur de renseignement" contiendra :

- pour chaque j l'adresse minimum de Tj c'est à dire Tj $[a_1, a_2 \ldots a_n]$ sur 5 positions et le nombre d'identificateurs qui le précédent dans la section, c'est à dire j (sur 2 positions) ($j \leqslant 99$)
- l'indicatif sur 1 position
- la dimension du tableau sur 2 positions

- a_1
- c_1
- ⋮
- a_n
- c_n

5 positions chacun

Ce "vecteur de renseignementé peut être schématisé aussi :

Il occupe 7 p + 10 N+3 positions.

3. 4. Zone réservée aux composantes d'un tableau

A l'exécution sera réservée une zone de mémoire où seules les composantes des tableaux seront rangées.

Les zones attribuées aux divers tableaux seront réservées par adresses décroissantes. Mais pour chaque tableau, les composantes seront rangées par

adresses croissantes (voir paragraphe 4) : la variable indicée T [a_1, a_2, ... a_n] occupera les positions de mémoire dont l'adresse est la plus petite ; nous appelons adresse minimum d'un tableau l'adresse la plus basse de la zone qui lui est réservée.

Soit D une section de tableau, qui n'est pas la première du programme. Nous dirons qu'une section de tableau D' précéde D si D' et D se trouvent en tête du même bloc et D' est écrite avant D, ou si le bloc en tête duquel se trouve D', contient strictement le bloc en tête duquel se trouve D.

Au moment où on rencontre (à l'exécution) la section D, les seuls tableaux dont on puisse avoir besoin en mémoire sont ceux dont la section précéde D. Au moment où on compile D, les adresses des "vecteurs de renseignements" des tableaux dont la section précéde D se trouvent dans la table. Il suffit de chercher la dernière de celles qui ont été introduites dans la table : on connait aussi l'adresse T ou se trouvera l'adresse minimum A0 du dernier tableau dont la section précéde D : les tableaux de D seront rangés à partir de A0.

Pour chacun on calcule sa longueur qui est égale à $(C_1 \times C_2 \times \ldots C_n)\,k$ avec k = 1 pour un type booléen, k = 12, pour un type entier ou réel. Dans son adresse minimum

Le principal avantage de cette méthode est qu'aucune instruction ne sera effectuée à la sortie d'un bloc à l'exécution. En particulier ce traitement évitera des difficultés lors des sorties de bloc par un saut (ALLERA).

Cas des déclarations de tableau se trouvant dans un corps de procédure.

La méthode précédente ne s'applique pas, car théoriquement le corps de procédure doit être "recopié" à l'emplacement de chaque appel [7].

Exemple :

```
DEBUT TABLEAU T1 [0 : 1] ;
    PROCEDURE F (X) ;
        DEBUT TABLEAU T2 [0 : 10];
                    :
        FIN
        DEBUT TABLEAU T3 [0 : 5] ;
                    :
                  F (A)
        FIN
FIN
```

A, ayant été déclaré dans un bloc plus externe.

Si on applique strictement la méthode précédente, les composantes de T3 et T2 seront sur la même zone, ce qui ne doit pas se produire. Il faut donc apporter quelques modifications.

A chaque exécution d'un appel de procédure la première adresse disponible à cet instant pour les tableaux sera cherchée dans la table.

Cette adresse sera la première adresse du premier tableau du corps de procédure. Plus précisément il sera effectué les opérations suivantes :

a) A chaque appel de procédure rencontré en compilation, on cherchera l'adresse de l'adresse minimum utilisée pour ranger les tableaux :

cette recherche est faite comme pour une déclaration de tableau (cf b). Un ordre sera construit [6] qui rangera cette adresse sur une pile, avec les autres renseignements nécessaires à la procédure (adresse de retour; index, lien).

b) A chaque déclaration de tableau rencontré en compilation ;

On remonte la table des identificateurs jusqu'à ce que l'on trouve :

soit α) un identificateur de tableau : cas normal

soit β) un identificateur de procédure :

Alors un test est fait sur le niveau p de cette procédure :

(β_1) Le tableau étudié est hors de cette procédure (niveau actuel de procédure $< p$) la consultation de table est continuée

(β_2) Le tableau étudié est dans la procédure (niveau actuel de procédure $= p$)

L'adresse minimum nécessaire sera donc une pile grâce à a) et c'est là qu'elle sera prise.

En réalité [6] n'effectuera pas l'opération décrite au a) pour les appels d'une procédure dont on est sûr que son corps ne contient aucune déclaration de tableau : il s'agit des appels situés aprés la fin de la déclaration d'une telle procédure.

4. CALCUL DE L'ADRESSE D'UNE VARIABLE INDICEE :

Le calcul d'adresse d'une variable indicée se fera à l'exécution, puisque les indices peuvent être des expressions arithmétiques, et les renseignements contenus dans le "vecteur de renseignement" nécessaire à ce calcul d'adresse ne sont connus qu'à l'exécution.

Méthode de rangement :

A partir de l'adresse minimum du tableau les composantes seront rangées en séquence de la façon suivante :

$T[a_1, a_2 \dots a_{n-1}, a_n]$, $T[a_1, a_2, \dots a_{n-1}, a_n+1] \dots$, $T[a_1, a_2 \dots, a_{n-1}, b_n]$

$T[a_1, a_2 \dots a_{n-1}+1, a_n]$, $T[a_1, a_2 \dots a_{n-1}+1, a_n+1] \dots$ $T[b_1, b_2, \dots, b_n]$.

C'est le dernier indice qui variera le premier de a_n à b_n compris. Dans le cas d'un tableau de dimension 2 (matrice), $T[I, J]$, si I est l'indice de la matrice, sera rangé ligne par ligne.

Définition du rang

Soit une variable indicée : $T[I_1, I_2 \dots I_n]$, $T[0, 0, 0 \dots 0]$ aura pour rang zéro par définition.

Soit alors $\alpha_0, \alpha_1 \dots \alpha_k$ la suite des éléments rangés comme il a été dit, r sera le rang de α_r.

a) Calcul du rang R de la composante $T[J_1, J_2, \dots, J_n]$ lorsque $a_1 = a_i = \dots a_n = 0$, par récurrence sur n :

$n = 1 \quad R_1 = J_1$

$n = 2 \quad R_2 = J_1 C_2 + J_2$

$n = 3 \quad R_3 = R_2 C_3 + J_3$

Formule de récurrence :

$$\boxed{R_n = R_{n-1} C_n + J_n} \qquad \text{avec } R_0 = 0,$$

Car chaque élément $T[J_1, J_2 \dots J_{n-1}]$ d'un tableau dimension n-1 donne naissance à C_n éléments $T1[J_1, J_2 \dots J_n]$ d'un tableau de dimension n, que l'on pourrait écrire $T[J_1, J_2 \dots J_{n-1}][J_n]$.

b) Cas général

Soit la déclaration de tableau suivante :

TABLEAU $T[a_1 : b_1, a_2 : b_2 \dots a_n : b_n]$;

$$Ck = b_k - a_{k+1}$$

Calculons le rang de l'élément $T[I_1, I_2, \dots I_n]$; on se raméne au cas a) en posant $J_k = I_k - a_k$,

donc :

$$R_1 = I_1 - a_1$$

$$R_2 = R_1 C_2 + I_2 - a_2$$

$$R_3 = R_2 C_3 + I_3 - a_3$$

$$\boxed{R_n = R_{n-1} C_n + I_n - a_n} \quad (1)$$

Alors si l'adresse de $T[a_1, a_2, \dots a_n]$ est A_0 ("adresse minimum") celle de $T[I_1, I_2 \dots I_n]$ est :

$$\boxed{A_0 + R_n}$$

Remarque : on a aussi en transformant la formule :

$$R_n = \sum_{k=1}^{N} C_k (I_k - a_k)$$

avec $$C_k = \prod_{r=k+1}^{n} (b_r - a_r + 1)$$

Le calcul du rang sera donc extrêmement simple si l'on connait les k C k, et a k. Il est effectué en utilisant la formule de récurrence (1).

Les C_k et a_k seront des renseignements gardés à la déclaration dans le "vecteur de renseignement" dans le but de ce calcul de rang. I_k sera calculé à l'exécution et perdu sitôt utilisé. On vérifiera que $\boxed{a_n \leq I_n \leq b_n}$ comme b_n n'est pas gardé dans le "vecteur de renseignement", pratiquement on vérifiera que : $o \leq I_n - a_n < C_n$

puisque $C_n = b_n - a_n + 1$

5. REALISATION PRATIQUE

5.1. Compilation d'une déclaration de tableau

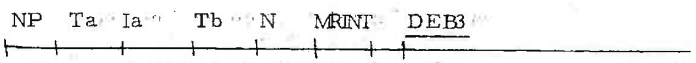
A la compilation d'une section de tableaux, deux choses seront faites :

- Formation de la table des symboles.
- Création d'ordres pour l'exécution.

Ceci sera fait en réalité, simultanément.

a) Formation de la table des symboles : l'adresse du"vecteur de renseignement"prise sur (MRINT) sera mise en face de l'identificateur de tableau, avec le type (1 position) et la dimension (2 positions).

b) Création d'ordres : Ces ordres permettront à un sous-programme d'exécution de construire le"vecteur de renseignement"se rapportant à la section de tableau étudiée. Les renseignements nécessaires à ce sous-programme seront placés en tête du sous-programme (cf [8]), de la façon suivante :



Sous-programme de Construction d'un "vecteur de renseignement".

MRINT : adresse où est stockée la valeur numérique de la 1° borne de la section des tableaux. Toutes les autres bornes seront stockées en séquence aprés cette première adresse, leur valeur étant entière et sur 12 positions, qui sera réduite à cinq dans le "vecteur de renseignement", si cela est possible.

N : dimension des tableaux sur 2 positions.

Tb : adresse du "vecteur de renseignement" précédent immédiatement dans la table, la section des tableaux étudiée. (5 positions).

Ta : adresse du "vecteur de renseignement" de la section de tableaux étudiée.

I_a : indicatif de type des tableaux étudiés.

NP : nombre d'identificateurs de tableaux, déclarés dans la section de tableau.

Enoncé des Σ et Σ' pour une section de Tableaux.

a) Σ'(TABLEAU,) Σ'(TABLEAU[) référencés (TBLEA). Ces deux Σ' ont même référence, pour éviter un ordre de branchement.

Σ'(TABLEAU,) : Il compte le nombre de virgules dans la section de tableau, ce calcul est fait dans NP. Chaque virgule est supprimée à la fin de ce Σ'.

Σ'(TABLEAU[) : Il garde sur (MRINT), 7xNP positions pour les NP identificateurs de la section de tableaux.

Il recherche dans la table des symboles, l'adresse du "vecteur de renseignement" précédent la section de tableaux étudiée, c'est à dire Tb. Cette recherche se fait par un sous-programme référencé (PØRTE) qui tient compte du cas ou nous sommes dans une procédure,

Dans le cas normal Tb est mis sur S (V) avec son type. Dans le cas d'une procédure, l'adresse mise sur S (V) (TABLØ) est une adresse donnée par [6] qui contiendra à l'exécution un Tb, différent du précédent, mais qui sera le bon.

Si nous sommes au premier tableau déclaré, il sera mis sur S (V), l'adresse d'initialisation (INI) de la zone où seront stockées les composantes des tableaux.

On sait que les bornes de la section de tableaux, devront avoir été déclarées dans un bloc plus grand. Pour vérifier ceci, à chaque entrée de bloc (CPTR2) est gardé dans (CTAB1). A chaque appel de bornes de la section de tableaux, (CPTR3), (qui est l'adresse dans la table de l'identificateur correspondant à la borne), sera comparé à (CTAB1).

Si CPTR3 > CTAB1 , alors il y a erreur, puisque la borne est déclarée dans le même bloc que la section de tableaux. ERR5 est frappé à la machine à écrire.

Exemple :

```
DEBUT  ENTIER  I,J,K,A ;
       :
       DEBUT  REEL A ; TABLEAU T[I : A, J : K];
              :
       FIN
       :
FIN
```

Dans ce cas il y a bien erreur de programmation, et l'erreur est décelée.

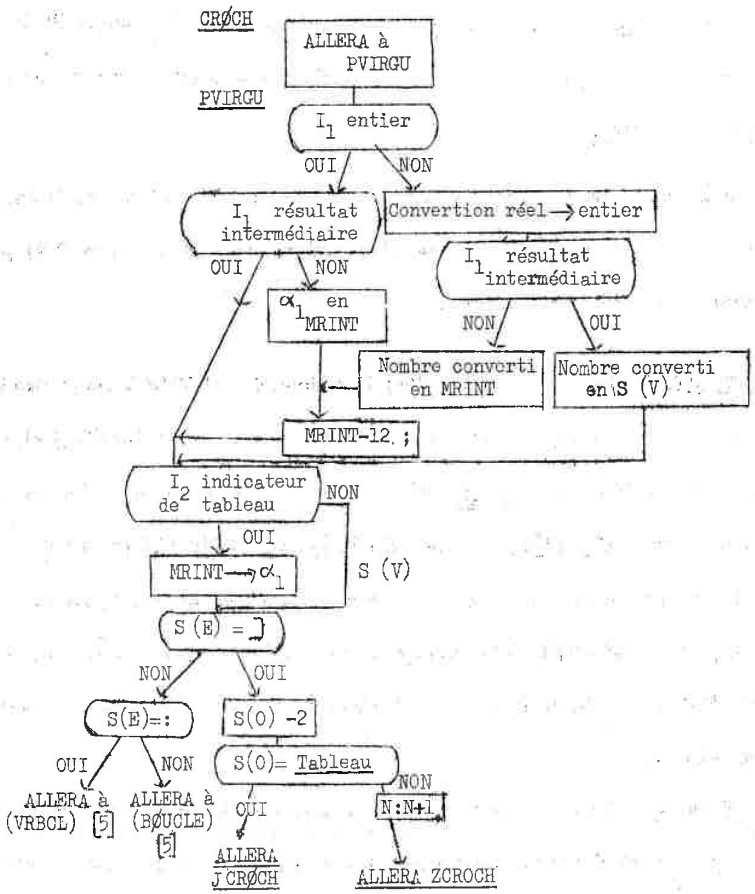
Remarque : Si A avait été déclaré, après la déclaration de tableau: l'erreur n'aurait pu être trouvée.

b) $\Sigma'(\sqsubset :)$ et $\Sigma'(\sqsubset ,)$ référencés (PVIRGU) (organigramme 1) $\Sigma'(\sqsubset ,)$ est utilisé deux fois : à la déclaration, et au calcul d'adresse d'une variable indicée (voir plus bas) ; seule la sortie est différente dans les deux cas. Cette différence sera faite en testant si en dessous du sommet S (O) de la pile O il y a TABLEAU ou non.

Ce Σ' stocké en séquence les bornes ou les indices, entre les deux crochets d'une déclaration de tableau, ou d'une variable indicée, en les convertissant en entiers s'ils sont réels.

On soustrait 12 à (MRINT) à chaque stockage, si un indice ou une borne n'est pas un résultat partiel. Pour pouvoir réinitialiser (MRINT) à la fin de l'utilisation de ce stockage, (MRINT) correspondant à la 1° borne ou au 1° indice est gardé sur S (V) en testant si $\alpha_2 = 5, \bar{5}$ ou $\bar{6}$. C'est ce (MRINT) qui sera envoyé en (DEB3-1) (chap II 5.1).

ORDINOGRAMME 1



Dans le cas du calcul d'adresse de la variable indicée, le nombre d'indices est compté en (N) afin de pouvoir le comparer à N dimension du tableau correspondant, il devra y avoir égalité, sinon ERR4 sera frappé à la machine à écrire.

c) Σ (:) référencé (DEPOIN) Il compte les : à la déclaration d'une section de tableaux pour connaitre sa dimension, ceci est fait dans (N) qui sera finalement mis dans la table.

d) Σ_1 ([) référencé (CRØCH) ; il se branche à PVIRGU pour stocker la dernière borne ou le dernier indice. On en ressort par (JCRØCH) si en dessous de S (O) il y a TABLEAU, c'est à dire dans le cas de la déclaration.

Donc en fait Σ_1 ([), comprend Σ'([,) et (JCRØCH).(JCRØCH) construit les ordres qui vont envoyer les renseignements devant le sous-programme de construction d'un vecteur de renseignement. Ces ordres sont sous forme d'une constante (Z4) qui est envoyée en (MPGØB) (chapitre I, programme objet).

En S (V) se trouve l'adresse de la table où doit être mise l'adresse du vecteur de renseignement, prise sur (MRINT), ceci est aussi fait par (JCRØCH) ; et 10 (N+1) positions sont réservées sur (MRINT) pour le vecteur de renseignement. Lorsque cette séquence est terminée,] doit supprimer [, comme dans le cas des paranthèses ([5]). On se branche allors (SYMBØL) ([5]).

e) Σ (TABLEAU) référencé TAB2. Ce Σ termine la déclaration de tableau, en supprimant TABLEAU sur S (O) et le type s'il éxiste. NP est remis à zéro.

Sous-programme de construction d'un vecteur de renseignement.

Ce sous-programme est effectué à l'exécution. Le vecteur de renseignement étant construit suivant la forme définie au paragraphe 3.

La référence de ce sous-programme est DEB3. Il met en séquence dans la zone réservée sur MRINT à la compilation, l'indicatif de type, la dimension, a_1, c_1, ... a_n, c_n. En sachant qu' en (DEB3-1) il y a l'adresse de la zone ou sont stockées de 12 en 12 positions : a_1, b_1 ... a_n, b_n.

En pratique, il sera gardé pour chaque a k et c k 5 positions.

Donc :

$|a\ k| < 10^5$ sinon ERR13 est frappé

$|c\ k| < 10^5$ sinon ERR14 est frappé

$c_n > 0$ sinon ERR15 est frappé.

On calcule aussi $(C_1 \times C_2 \times \ldots C_n) \times k$ appelé longueur du tableau

où : $k = 1$ si $I_a = \bar{6}$

$k = 12$ si $I_a = 5$ ou $\bar{5}$.

et l'adresse minimum du tableau sera l'adresse minimum du tableau précédent (Tb) diminuée de la longueur du tableau étudié.

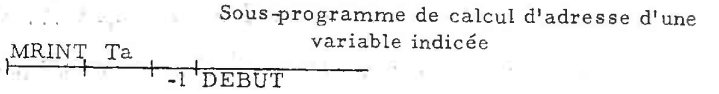
S'il y a p identificateurs de tableau dans la section de tableau étudiée, l'adresse minimum de la zone réservée à la section sera égale à l'adresse minimum de Tb diminuée de p fois la longueur, commune à tous les tableaux.

Cette adresse sera bien la dernière adresse minimum mise dans la table par (JCRØCH), c'est à dire celle qui sera nécessaire dans la recherche de Tb (pour la délcaration de section de tableaux suivante,) par le sous-programme (PØRTE) à la compilation.

5. 2. Compilation d'une variable indicée

On doit construire les ordres, qui donneront les renseignements nécessaires au calcul de l'adresse de la variable indicée.

Ces renseignements peuvent être schématisés en Mémoire de la façon suivante :



Ta : adresse du "vecteur de renseignement" correspondant à l'identificateur de la variable indicée.

MRINT: adresse où est stockée en valeur entière le premier indice, les autres étant placés en séquence de 12 en 12 positions suivant (MRINT).

Enoncé des Σ et Σ' de la compilation d'une variable indicée :

a) Σ_2 ([). Il commence en (CRØCH) va à (PVIRGU) comme Σ_1 ([), la différence se fait comme il a été déjà dit (Si en dessous de S (O) il y a TABLEAU). Dans ce cas il n'y aura jamais TABLEAU, on ira donc à (ZCRØCH).

Les ordres sont sous forme d'une constante (Z3) qui sera mise en (MPGØB) (chapitre I, programme objet). (MRINT) sera réinitialisé grâce à

MRINT correspondant au premier indice, et gardé sur S (V) par (PVIRGU).

Sur S (V) sera mis (MRINT) avec "flag". C'est là qu'à l'exécution sera mis le calcul d'adresse de la variable indicée. La composante correspondant à cette variable indicée aura un indicatif de type différent, suivant celui de la variable indicée :

Si $I_2 = \overline{5}$ on aura $I_2 = 0$, pour indiquer que la composante est réelle.

Si $I_2 = 5$ on aura $I_2 = 1$, pour indiquer que la composante est entière.

Si $I_2 = \overline{6}$ on aura $I_2 = 3$, pour indiquer que la composante est booléenne.

C'est cette nouvelle valeur de I_2 qui sera mise sur S (V). Dans le cas d'un paramétre formel spécifié Tableau ([6]), la première fois qu'il sera rencontré le nombre d'indices comptés en (N) sera mis dans la table, dans la place correspondant à la dimension d'un tableau. Les autres fois (N) sera comparé à cette valeur mise dans la table, il devra y avoir égalité, sinon ERR4 sera frappé.

Sous-programme de calcul d'adresse d'une variable indicée :

Ce sous-programme utilise la formule :

$$\ell_n = R_{n-1} \times C_n + I_n - a_n$$

$a_1, c_1, \ldots a_n, C_n$ sont donnés par le vecteur de renseignement

$I_1, I_2 \ldots I_n$ sont stockés en séquence à partir de MRINT (en DEBUT-6).

Donc ce sous-programme référencé DEBUT sera extrêmement simple. Il comprendra en outre deux tests d'erreurs.

- Ik - ak $\geq$ 0 , sinon ERR11 est frappé

- Ik - ak $<$ Ck, sinon ERR12 est frappé.

Le résultat du calcul d'adresse sera mis sur (MRINT) à l'adresse ou était stockée la valeur de la première borne (est mise sur S (V)) à la compilation). L'adresse calculée de la variable indicée sera finalement

$$A_n = AMIN + R_n$$

où AMIN est l'"adresse minimum" donnée par le "vecteur de renseignement".

CHAPITRE III

TABLEAUX REMANENTS

1. CONSERVATION DES COMPOSANTES DU TABLEAU

La valeur d'une variable rémanente ne doit pas changer entre la sortie d'un bloc et la rentrée suivante.

Pour les variables simples déclarées rémanentes ceci est résolu simplement en leur donnant comme adresse, une adresse prise dans une portion de mémoire qui ne sera jamais réinitialisée ou perdue (REMAN).

Une nouvelle zone sera gardée à l'exécution pour le rangement des composantes des tableaux, cette zone sera indépendante de celle réservée aux composantes des tableaux locaux. Elle sera gérée (à l'exécution) par un compteur d'affectation "d'adresse minimum" (AMIN I) initialisé par le programme objet. Les zones nécessaires aux tableaux successifs seront réservées par adresses décroissantes.

2. CONSERVATION DU VECTEUR DE RENSEIGNEMENT

Le "vecteur de renseignement" doit être gardé, pour faire le calcul d'adresse d'une variable indicée, à l'intérieur du même bloc. Mais aussi parcequ'à une nouvelle entrée dans ce bloc les bornes du tableau pourront varier, aussi il faudra comparer les nouvelles bornes aux anciennes.

Donc "le vecteur de renseignement" d'un tableau se trouvera toujours dans la même zone, prise dans REMAN. La valeur des bornes seule pourra changer à chaque passage, à l'exécution, par la déclaration de la section de tableau.

3. DIFFERENCE ENTRE LES DIVERS PASSAGES PAR UNE DECLARAT

DE TABLEAU REMANENT.

Au premier passage, on se borne à construire le "vecteur de renseigne et à réserver une zone de stockage à la suite des tableaux déjà rangés.

Aux passages suivants, il peut y avoir modification des bornes, ce qui ob ge à modifier l'étendue de la zone du stockage, et à réarranger les composantes d la nouvelle zone.

Un tableau représentera la valeur que prennent les variables indicées ent une déclaration de section de tableau et la suivante.

Ancien tableau	Nouveau tableau	
	indice indéterminé	indice indéfini
indice indéfini	valeur indéfinie	valeur indéfinie
indice défini	valeur perdue	valeur gardée

On distingue trois cas :

α) si une suite d'indices est définie à la fois pour l'ancien et le nouveau tableau, la valeur de la variable indicée correspondante doit rester inchangée.

β) si une suite d'indices définie pour l'ancien tableau ne l'est plus pour le nouveau, la valeur de la variable indicée correspondante est perdue.

γ) si une suite d'indices, qui était indéfinie pour l'ancien tableau devient définie pour le nouveau, la valeur de la variable indicée correspondante est indéfinie, mais un espace doit être gardé dans le nouveau tableau pour qu'une valeur puisse lui être affectée.

4. CONSTRUCTION ET MODIFICATION DU "VECTEUR DE RENSEIGNEMENT"

a) premier passage par une déclaration.

Le "vecteur de renseignement" d'une section de tableau rémanente à la même structure que celle décrite au chapitre II. Il est construit sur (REMAN)

L'"adresse minimum" est obtenue en soustrayant à (AMIN I) la longueur du tableau.

La position du "vecteur de renseignement" utilisée pour garder le type de la section de tableau, sera utilisée, pour faire la différence entre la première déclaration, et les autres.

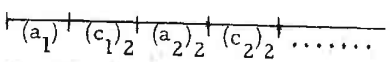
- Au premier passage, il y aura zéro, dans cette position.
- Aux autres passages, il y aura le digit du type.

Ainsi le test de premier passage, pourra être fait aisément.

b) Autres passages.

Les bornes peuvent changer de valeur (la dimension et le type sont invariants). Pour reconnaître si un nouvel arrangement des composantes est nécessaire, il va falloir comparer la valeur des nouvelles bornes à celle des anciennes.

Pour cela on construira un pseudo "vecteur de renseignement" contenant les nouvelles valeurs $(a_i)_2$ des bornes inférieures et $(c_i)_2$ des longueurs de variation des indices (cf figure).



Après la comparaison, on transférera ce pseudo "vecteur de renseignem dans la partie correspondante du "vecteur de renseignement".

Le pseudo-vecteur pourra alors être perdu : on le construira donc sur (MRINT).

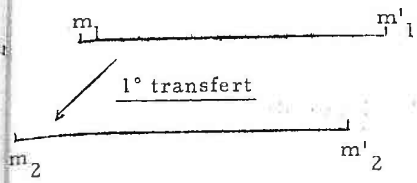
5. MODIFICATION DE LA ZONE DE STOCKAGE ET REARRANGEMENT

- Si toutes les bornes sont égales rien n'est fait.

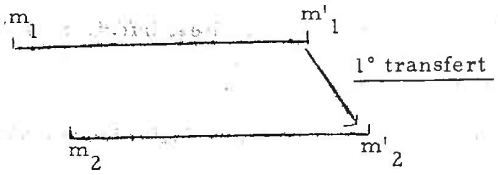
-- Si une des bornes prend une valeur différente, il faudra effectuer une nouvelle réservation, et un réarrangement des composantes de la section tableau. Ces deux opérations nécessitent des transferts d'une partie de la mémoire formée des positions dont les adresses sont comprises entre m_1 et m'_1 dans une autre formée des positions comprises entre m_2 et m'_2.

$(m_1 - m'_1 = m_2 - m_2)$

Notons que lorsque ces parties de mémoire se chevauchent, pour éviter toute destruction de positions à transférer, si $m_2 < m_1$ (autrement dit $m'_1 < m'_2$ le transfert devra commencer par la position de plus faible adresse (m_1) : nous disons que c'est un transfert par adresses croissantes ; si $m_2 > m_1$, on effectura au contraire un transfert par adresses décroissantes.



Transfert pas adresses croissantes.



Transfert pas adresses décroissantes

Nous traiterons d'abord le cas des tableaux à une dimension puis nous y raménerons le cas général.

5. 1. Nouvelle réservation

Soit e_1 la longueur de l'ancien tableau, et e_2 la longueur du nouveau tableau.

α) $e_1 > e_2$

La longueur du tableau diminue, donc il faudra "tasser" les tableaux qui suivent celui qui est étudié, pour pouvoir réutiliser l'espace libéré. Ce transfert n'est pas fait immédiatement car il pourrait faire perdre des composantes de l'ancien tableau, nécessaires au nouveau. Il sera effectué après réarrangement des composantes.

Les "adresses minima" des deux tableaux restent donc les mêmes pour l'instant.

β) $e_1 = e_2$

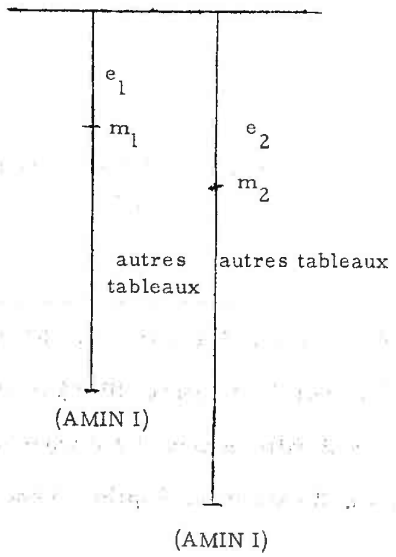
Aucune réservation nouvelle n'est à prévoir.

γ) $e_2 > e_1$

La longueur du tableau augmente donc il faut prévoir un nouvel espac pour réarranger les composantes.

La nouvelle "adresse minima" du tableau est m_2 elle est inférieure à l'ancien adresse minimum m_1, nous avons $m_2 = m_1 + e_1 - e_2$.

Un transfert des tableaux suivants est effectué, il s'agit d'un transfert par adresses croissantes.



(AMIN I)

5, 2 Réarrangement des composantes du tableau

Dans chacun des cas étudiés au paragraphe précédent, la nouvelle adresse minimum m_2 est inférieure ou égale à l'ancienne m_1.

Soit A 1, B 1, les valeurs de la paire de bornes de l'ancien tableau.

et A 2, B 2, celles du nouveau tableau.

C 1 et C 2 la longueur de variation des indices.

Si $A1 > B2$ ou si $A2 > B1$, aucune composante ne sera commune aux deux tableaux. Aucun réarrangement n'est nécessaire. En fait puisque B 2 et B 1, ne sont pas utilisés par le "vecteur de renseignement", les tests effectués seront :

si $A2 > A1$, $A2 > C1 + A1 - 1$ ou plus simplement $A2 - A1 \geqslant C1$

si $A2 < A1$, $A1 > C2 + A2 - 1$ soit $A1 - A2 \geqslant C2$

Plaçons-nous dans le cas où les deux tableaux ont des composantes en commun.

Soit $M = A2 - A1$

1°) $M \geqslant 0$

La partie de l'ancien tableau formée des composantes dont l'indice est compris entre A 2 et B 1 contient les composantes communes aux deux tableaux. Elle sera transférée à partir de la nouvelle position origine m_2.

Comme $m_2 \leq m_1 \leqslant m_1 + M$,

il s'agit d'un trasnfert par adresses croissantes, de longueur C 1 - M (certaines composantes n'appartenant pas au nouveau tableau seront inutilement transférées lorsque $B2 < B1$)

2°) $M < o$

La première position à transférer est la position d'adresse m_1 qui doit venir en $m_2 - M$

a) si $m_2 - M \leqslant m_1$

Soit $A\,1 - A\,2 \leqslant C\,2 - C\,1$ ou encore $B\,1 \leqslant B\,2$, on se ramène alors au premier cas, c'est à dire un transfert par adresses croissantes. Cette fois des composantes seront transférées inutilement mais cela simplifie l'étude.

b) $A\,1 - A\,2 > C\,2 - C\,1$ ou encore $B\,2 > B\,1$

On transférera par adresses décroissantes les composantes de l'ancien tableau dont les indices sont compris entre A 1 et B 2, la composante d'indice B 2 vient remplacer la composante d'indices B 1 à l'adresse $m_1 + C\,1 = m_2 + C\,2$. La longueur du transfert est $C\,2 - |M|$.

Dans le cas ou $C\,2 < C\,1$ nous pourrons après ce réarrangement des composantes du tableau, décaler le tableau étudié et les tableaux suivants pour occuper les positions de mémoire laissées libres. Il s'agira d'un transfert par adresses décroissantes.

5. 3. Généralisation à un tableau de dimension quelconque

Soit d'abord un tableau de dimension 2, dont les bornes inférieures ont pour valeurs $(a_1)_i$ et $(a_2)_i$, et les bornes supérieures $(b_1)_i$ et $(b_2)_i$, l'indice i valant 1 pour l'état "ancien" et 2 pour l'état "nouveau" ; soit aussi

$$(c_1)_i = (b_1)_i - (a_1)_i + 1$$

$$(c_2)_i = (b_2)_i - (a_2)_i + 1$$

les longueurs de variation des différents indices.

Considérons l'ancien tableau comme formé des $(c_1)_1$ tableaux de dimension 1, dont les indices varient de $(a_2)_1$ à $(b_2)_1$; réarrangeons chacun de ces tableaux comme il a été dit en $(c_1)_1$ tableaux de dimensions 1 dont les indices varient de $(a_2)_2$ à $(b_2)_2$: ils constituent alors un tableau de dimension 1 de longueur $(c_1)_1 \times (c_2)_2$, dont on peut nommer la borne inférieure $(a_1)_1 \times (c_2)_2$ et qui doit être remplacé par un tableau de dimension 1, ayant pour borne inférieure $(a_2)_2 \times (c_2)_2$ et pour longueur $(c_1)_2 \times (c_2)_2$.

Exemple

$(a_1)_1 = 0$ $(b_1)_1 = 2$ $(c_1)_1 = 3$

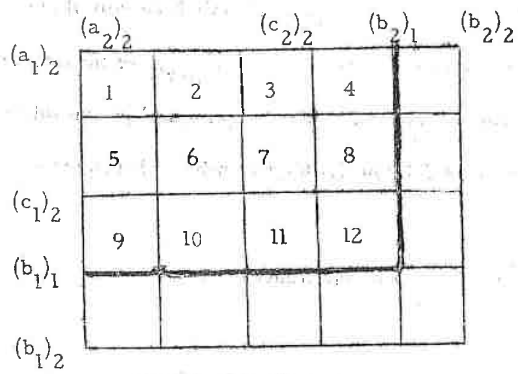
$(a_2)_1 = 1$ $(b_2)_1 = 4$ $(c_2)_1 = 4$

$(a_1)_2 = 0$ $(b_1)_2 = 3$ $(c_1)_2 = 4$

$(a_2)_2 = 1$ $(b_2)_2 = 5$ $(c_2)_2 = 5$

Soit schématiquement :

a)



En mémoire nous avons successivement :

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

- 1 2 3 4 b 5 6 7 8 b 9 10 11 12 b

- 1 2 3 4 b 5 6 7 8 b 9 10 11 12 b b b b b.

b signifiant blanc ou tout autre chose

Si le tableau et de dimension 3, nous faisons le même processus en commençant par étudier $(c_1)_1 \times (c_2)_1$ tableaux à 1 dimension qui consituera alors un tableau de longueur $(c_1)_1 \times (c_2)_1 \times (c_3)_2$. Et on continue.

Soit maintenant un tableau de dimension n, $(a_r)_1$, $(c_r)_1$, les valeurs la $r^{è}$ borne inférieure et de longueur de variation du $r^{è}$ indice dans l'état "ancien" $(a_r)_2$ et $(c_r)_2$ dans l'état "nouveau" (r= 1,2 ... n).

Supposons chaque tableau partiel de dimension r obtenu en fixant n-r premiers indices ($r < n$) rangé, pour les nouvelles valeurs des bornes, en un tableau à une dimension de longueur e_r.

On considère chaque tableau partiel de dimension r+1, obtenu en fixant les n-r-1 premiers indices comme un tableau à une dimension de borne inférieure $e_r \times (a_{n-r})_1$ et de longueur $e_r \times (c_{n-r})_1$, qui doit être remplacée par un tableau à une dimension de borne inférieure $e_r \times (a_{n-r})_2$ et de longueur $e_{r+1} = e_2 \times (c_{n-r})_2$; on doit effectuer ce réarrangement dans un tableau de dimension 1. Ainsi on parvient en faisant évoluer l'entier r à réarranger le tableau donné.

Notons qu'en particulier on effectue une nouvelle étude de la réservation pour chaque r et que :

Chaque tableau d'une section de tableau, sera étudiée séparément.

6.) REAJUSTEMENT DES "ADRESSES MINIMA" DES TABLEAUX QUI SUIVENT LE TABLEAU ETUDIE.

L'"adresse minimum" de la zone réservée au tableau étudié est modifiée, mais il en est de même pour celles des tableaux qui le suivent en mémoire. Il faut donc les modifier dans les "vecteurs de renseignement." associés.

Remarquons que l'ordre de construction des "vecteurs de renseignement," sur (REMAN) des tableaux rémanents, n'est pas celui de l'écriture du programme ALGØL, mais celui du premier passage par leur déclaration à l'exécution ; les composantes des tableaux sont rangées dans ce même ordre.

A chaque premier passage par une déclaration, on gardera dans une partie connue de la mémoire l'adresse du "vecteur de renseignement" associé. Quand le réarrangement du tableau T étudié sera fini, on cherchera dans cette partie de la mémoire l'adresse du "vecteur de renseignement" du tableau T ; à partir de cette adresse (inclusivement) on ajoutera à toutes les "adresses minima" des "vecteurs de renseignement " la différence des longueurs du nouveau tableau et de l'ancien.

7. REALISATION PRATIQUE

7. 1. Indication dans la table qu'un tableau est rémanent

Comme nous l'avons dit, l'adresse du "vecteur de renseignement" sera prise sur (REMAN).

Dans le cas des tableaux locaux, la recherche de la section de tableau précédent celle étudiée, se faisant en testant sur le type de l'adresse mise dans le table. Mais comme ce type est le même pour les tableaux rémanents, on pourrait tomber sur l'un d'eux. Dans ce cas il faudra donner une indication supplémentaire

dans la table, pour éviter cette faute.

Cette indication sera donnée en plaçant un "digit" après le type du tableau dans la table. Il ne restera donc plus qu'une position pour la dimension du tableau. Donc dans le cas d'un compilateur acceptant les tableaux locaux et rémanents, la dimension des tableaux sera limitée à 9, mais il ne semble pas qu'on puisse utiliser des tableaux de dimension supérieure, même si il a été prévu 2 positions pour la dimension des tableaux locaux.

7. 2. Compilation Σ et Σ'.

Les Σ et Σ' obéissent aux mêmes régles que pour les tableaux locaux seuls, et les branchements seront aussi les mêmes.

Il faudra seulement tester à l'intérieur des Σ et Σ' si le tableau étudié est local ou rémanent.

Nous donnerons ici les différences, si elles existent, à l'intérieur des Σ et Σ'

- Σ'(TABLEAU) référencé (TBLEA) : identique

- Σ(TABLEAU [ ) référencé (TBLEA) si en dessous de S (0) il a REMANENT . Alors il n'y a pas de recherche de l'"adresse minimum" de la section de tableau, précédent celle étudiée. Sur S (V) est mis un indicatif 99999 à la place de cette adresse, il nous permettra à l'exécution de savoir que nous avons une déclaration de tableau rémanent.

- Σ'([:), Σ'([,) référencés (PVIRGU) : identiques.

- Σ (:) référencé (DEPOIN) : identique.

- Σ_1 ([ ) référencé (CRØCH), commence de la même façon, mais pour (JCRØCH) il y a certaines différences.

a) Mise d'un zéro dans la position réservée au type dans le "vecteur de renseignement" du tableau étudié.

Ce zéro sera mis une seule fois car l'ordre sera détruit en le transformant en un N Ø P (41). Aussi la première fois que le tableau sera déclaré il y aura 0 dans cette position, et toutes les autres fois, le type du tableau, qui est différent de 0.

- Σ_2 ([ ), identique.

- Σ (TABLEAU) référencé (TAB2) supprime sur S (O) REMANENT et le type, qui existe toujours.

7. 3. Sous-programmes d'exécution

Le sous-programme de calcul d'adresse est le même.

a) Sous-programme de construction d'un "vecteur de renseignement". Il sera le même pour la première déclaration d'une section de tableau. Mais construit sur (REMAN).

Le sous-programme de comparaison, de réservation, et de réarrangement, est donné sous-forme d'un ordinogramme détaillé (ordinogramme 2).

Le test C 2 - C 1 $\lessdot$ A 1 - A 2 sera remplacé par :

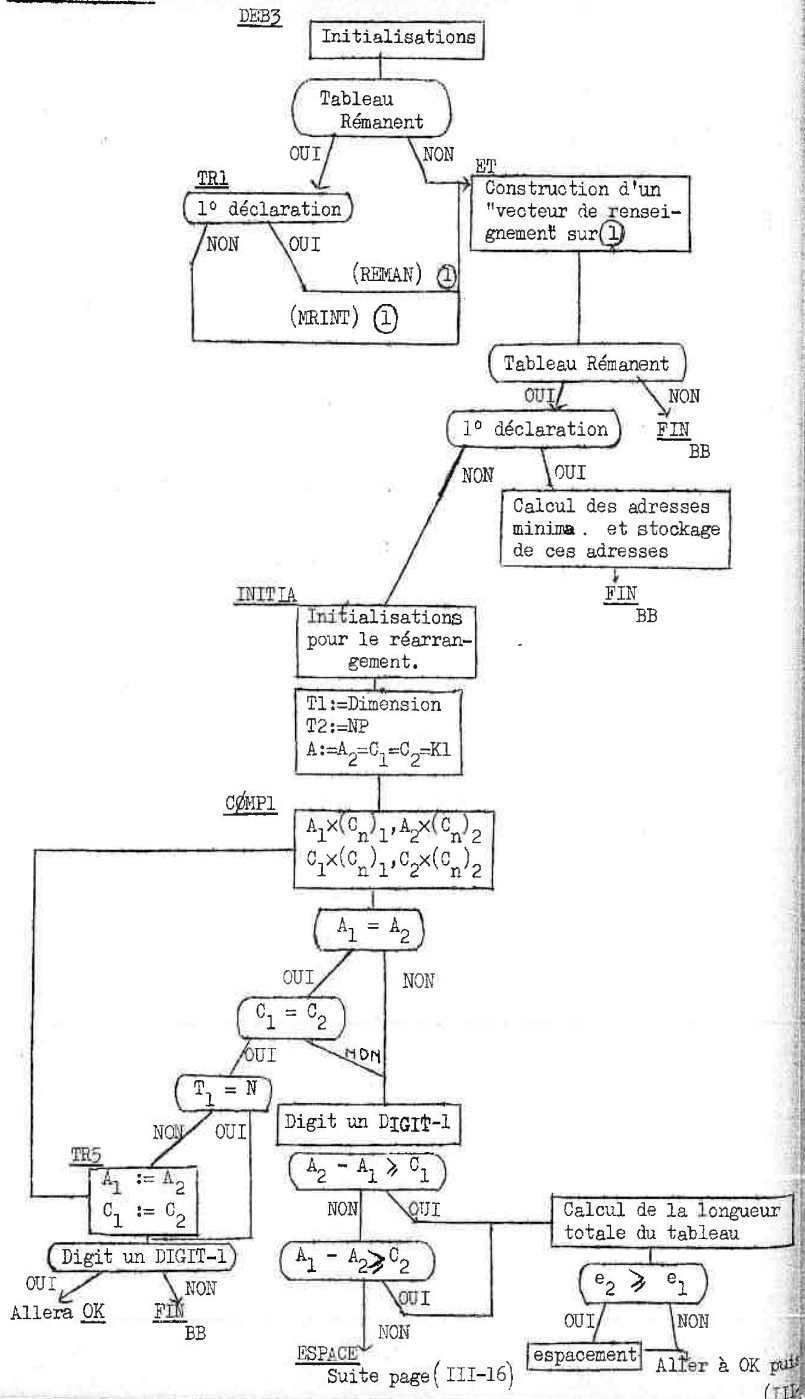
$m_2 - m_1 \lessdot$ A 1 - A 2

Car si C 2 $\geqslant$ C 1, $m_2 - m_1$ = C 2 - C 1 ; si C 2 $\lessdot$ C 1, $m_2 - m_1 = 0$ mais A 2 - A 1 = M est négatif.

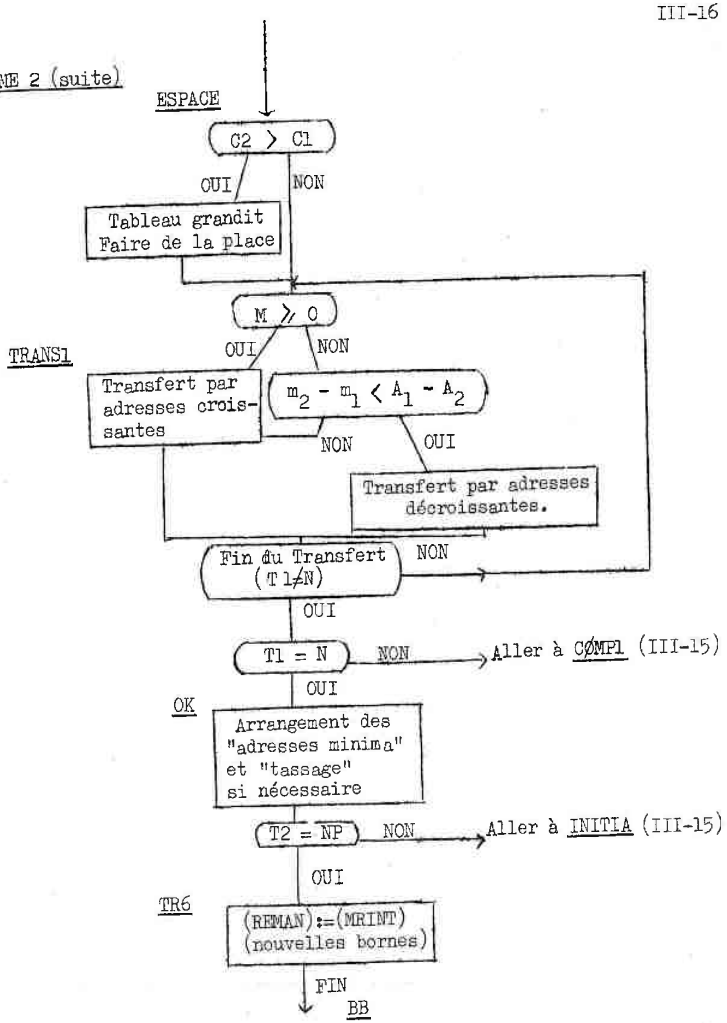
Le transfert pas adresses croissantes, se fera à l'aide d'un transfert d'enregistrement, qui est rapide, alors que le transfert par adresses décroissantes, se fera par des transferts de zone de longueur égale à la longueur

d'une composante de tableau soit 12 positions pour les composantes réelles et entières, et 1 position pour les composantes booléennes c'est-à-dire que l'on aura, soit un TFL (06). un TF (26) ou un TD (15), qui nous sera donné par le type du tableau. (notice S. P. S.)

ORDINOGRAMME 2



ORDINOGRAMME 2 (suite)



CHAPITRE IV

ALGOL MATRICIEL

1. GENERALITES

Nous nous proposons dans ce chapitre d'étendre ALGØL au calcul matriciel, en y introduisant :

a) des expressions matricielles où peuvent être effectuées des additions et soustractions, des multiplications de matrices, l'élévation d'une matrice à une puissance entière de signe quelconque (avec comme cas particulier l'inversion), la multiplication d'une matrice par un scalaire.

b) une instruction d'affectation d'une expression matricielle à une matrice.

2. SYNTAXE

Un seul symbole de base sera adjoint à ALGØL, pour désigner la transposition : on pourra le représenter par TR.

Nous ajouterons d'autre part à la grammaire définissant ALGØL (cf [6] , et [9]) les régles suivantes :

⟨expression matricielle simple⟩ ::= ⟨expression matricielle simple⟩

⟨opérateur additif⟩ ⟨terme matriciel⟩ | ⟨opérateur additif⟩

⟨terme matriciel⟩ | ⟨terme matriciel⟩

⟨terme matriciel⟩ ::= ⟨terme matriciel⟩ x ⟨facteur matriciel⟩

| ⟨facteur matriciel⟩ ⟨ terme⟩ x ⟨facteur matriciel⟩

⟨facteur matriciel⟩ ::= ⟨secondaire matriciel⟩ ↑ ⟨primaire arithmétique⟩

| ⟨secondaire matriciel⟩

⟨secondaire matriciel⟩ ::= TR ⟨primaire matriciel⟩ | ⟨primaire matriciel⟩

⟨primaire matriciel⟩ ::= ⟨identificateur de tableau⟩

| ⟨ (expression matricielle simple) ⟩

Pour l'affectation nous aurons :

⟨partie gauche⟩ ::= ⟨identificateur de tableau⟩ :=

⟨instruction d'affectation⟩ ::= ⟨liste de parties gauches⟩ | ⟨expression matricielle⟩

Dans la partie droite il ne doit y avoir que des identificateurs de tableau de même type.

Un identificateur de tableau constituant un primaire matriciel doit être déclaré de dimension 2.

3. SEMANTIQUE

Les opérations matricielles ont leur signification habituelle. On notera que, suivant les conventions mathématiques habituelles, le scalaire multipliant une matrice doit être placé devant elle. ↑ est utilisé pour l'élévation d'une matrice à une puissance entière : si l'expression arithmétique constituant cette puissance est de type réel, elle sera arrondie à l'entier le plus proche.

4. EXEMPLES :

- Expression matricielle

(A + B) x (C - D) + D ↑ (-1) + b x E,

A,B,C,D,E, étant des identificateurs de tableaux.

b un identificateur de variable simple entière.

- Affectation matricielle

A := (B + C) ↑ (X + Y) - E ;

A,B,C,E, étant des identificateurs de tableaux

X,Y des identificateurs de variable simple entière ou réelle.

- Programme matriciel

```
DEBUT TABLEAU A,B,C,D,E [0:5, 0:5] ; ENTIER I,J ;
   POUR I:=0 PAS 1 JUSQUA 5 FAIRE POUR J:=0
   PAS 1 JUSQUA 5 FAIRE A[I,J]:=B[I,J] := I+J ;
   C := D := E := (A + B) x B - A ↑ (-1)
FIN
```

5. MATRICE DE PRIORITE :

Il suffira d'ajouter à celle qui a été donnée par [5] une ligne et une colonne associées à T. R.

Un programme écrit par A. EMOND éléve au Centre de Calcul, sur la construction d'une matrice de priorité, à partir des régles de gramma ALGOL, et d'ALGØL matriciel donne les résultats suivants :

pour la ligne :	pour la colonne :
TR< (	+< TR
TR> ↑	+< TR
TR> +	-< TR
TR> -	:=< TR
TR> SINON	(< TR
TR>;	
TR> FIN	
TR>)	

6. PROBLEME DE RESERVATION

Les matrices déclarées par des tableaux, sont rangées à la place habituelle des tableaux. Mais on pourra avoir besoin de place en mémoire pour des résultats partiels.

Ces résultats seront placés dans la zone réservée aux tableaux : la première position sera donnée par l'adresse minimum de la dernière matrice résultat intermédiaire ou du dernier tableau déclaré. A chaque résultat intermédiaire matriciel il faudra donc associer, un "vecteur de renseignement" qui occupera 30 positions (N=2). Ce "vecteur de renseignement" sera appelé VRI (vecteur de renseignement intermédiaire), et sera construit sur (MRINT).

Cette zone de résultat intermédiaire pourra varier au cours des calculs, contrairement à la zone allouée aux tableaux qui restait fixe à l'intérieur d'un bloc. En effet un résultat intermédiaire pourra être perdu dés que le calcul intermédiaire sera opéré et la zone qui lui était allouée pourra être immédiatement réinitialisée.

7. SOUS-PROGRAMME D'EXECUTION

Ce sous-programme devra vérifier :

- que les tableaux sont déclarés à 2 dimensions.
- pour l'addition et la soustraction, que les deux matrices ont même nombre de lignes, et même nombre de colonnes.

- pour l'inversion, que la matrice est "carrée"

- pour la multiplication, que le nombre de colonnes de la première matrice, est égal au nombre de lignes de la seconde.

Lorsqu'une zone résultat intermédiaire sera nécessaire, un VRI devra être construit, qui contiendra l'"adresse minimum" calculée, de la zone de résultat intermédiaire, et les mêmes renseignements qu'un "vecteur de renseignement" de tableau.

Pour le calcul effectif des opérations matricielles, le programme a été inspiré d'un programme de l'EPL [10] (1620 / 5. 0. 22) qui a été modifié en tenant compte qu'en ALGØL, les matrices sont rangées ligne par ligne, au lieu de colonne par colonne, comme dans ce programme de l'E. P. L.

8. ORDRES A CREER POUR LE PROGRAMME OBJET

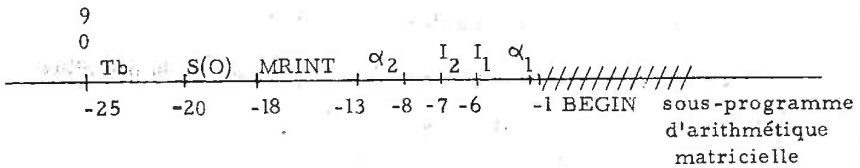
Une seule séquence de programme sera utilisée, pour créer des ordres envoyant les renseignements nécessaires à l'exécution. On se branchera à cette séquence, référencée (CAMATR) en testant 2 conditions.

- S (O) est un des symboles de base de l'ALGØL matriciel.

- sur S (V), en I_2 il y a un indicatif de tableau ($\bar{5}$, 5) ou de résultat partiel, qui sera 6.

Ces tests seront faits dans (GAMMA) de [5].

Les renseignements envoyés par l'ordre construit en (CAMATR) sont les suivants :



Où :

- α_1 I_1 ; est la première opérande et son type, sur S (V), si I_1 = 5 ou $\bar{5}$ α_1 indiquera l'adresse du "vecteur de renseignement" de la matrice étudiée. Si I_1 = 6 α_1 indiquera l'extrémité du VRI correspondant au résultat intermédiaire étudié.

- α_2 I_2 ; même chose, pour la 2° opérande, si elle éxiste.

- MRINT ; Dans le cas d'un nouveau VRI à construire (si I_1 et I_2 = 5 ou $\bar{5}$ MRINT indiquera l'extrémité de ce VRI.

- S (O) ; opérateur matriciel.

- Tb. Dans le cas ou nous avons besoin d'une zone de résultat intermédiaire, qui est la première, il faut connaitre la dernière adresse minimum de tableau, cette recherche se fera par le sous-programme (PØRTE), et sera mise en Tb.

- 9. Si nous avons besoin de Tb

- 0. Dans les autres cas, c'est-à-dire que la dernière adresse minimum utilisée sera dans VRI en MRINT - 30, L'ordinogramme ③ , donne les détails de toute cette construction. Aprés ceci on se branche à (PRIØR) [5] , qui testera la nouvelle priorité entre S (E) et S (O) qui est baissée dans (CAMATR).

9. REALISATION PRATIQUE

9. 1. Restrictions : par manque de temps les restrictions suivantes seront apportées à cet ALGOL matriciel.

- La transposition n'a pas été introduite.
- Seuls les opérations sur les matrices réelles peuvent être exécutées.
- Le sous-programme d'inversion n'a pu être écrit.

9. 2. Compilation : sur S (V) il y aura l'indicatif 6, chaque fois qu'une zone intermédiaire sera necéssaire.

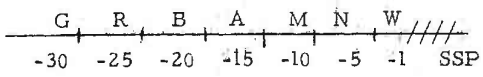
Les renseignements donnés à l'exécution, seront construits en (Z7). Chaque fois que dans α_1 I_1 il n'y aura pas l'indicatif, 6 ou 5, nous aurons besoin de la dernière "adresse minimum" de tableau, pour connaître la première position de zone intermédiaire, necessaire au calcul matriciel, cette recherche se fera par le sous-programme (PØRTE) de JCRØCH.

9. 3. Exécution : Lorsque I_1 = $\bar{5}$ et I_2 = 5 il faudra construire un VRI sur (MRINT). Les erreurs suivantes seront décalées :

- ERR16 : Matrice de dimension différente de 2.
- ERR17 : Dans une affectation, les matrices n'ont pas le même nombre de lignes ou de colonnes.
- ERR18 : Dans une addition, les matrices n'ont pas le même nombre de lignes ou de colonnes.
- ERR19 : Dans une multiplication, le nombre de colonnes de la première matrice, n'est pas égale au nombre de lignes de la seconde matrice.

Après ces calculs et ces recherches d'erreurs, nous nous brancherons au sous-programme de calcul effectif.

Les renseignements nécessaires à ce sous-programme seront :



A : adresse minimum de la première matrice, B de la seconde matrice, G de la matrice résultat, si A n'est pas une matrice résultat intermédiaire.

M : nombre de lignes d'une matrice, N ou R, son nombre de colonnes. Ce sous-programme se terminera par un BB c'est-à-dire que l'on continuera à effectuer le programme objet.

W : indicatif qui obéit aux régles suivantes :

W	Opération	Dimension de		
		A	B	C
0000	A + B → G	MxN	MxN	"
0001	A + B → A	"	"	"
0010	A - B → G	"	"	"
0011	A - B → A	"	"	"
0110	A → G	M N		M N
1000	A x B → G	M N	N, R	M, R
1001	b x A → A	M N	1 x 1	M N
0110	- A → G	M N		
0110	- A → A	"		

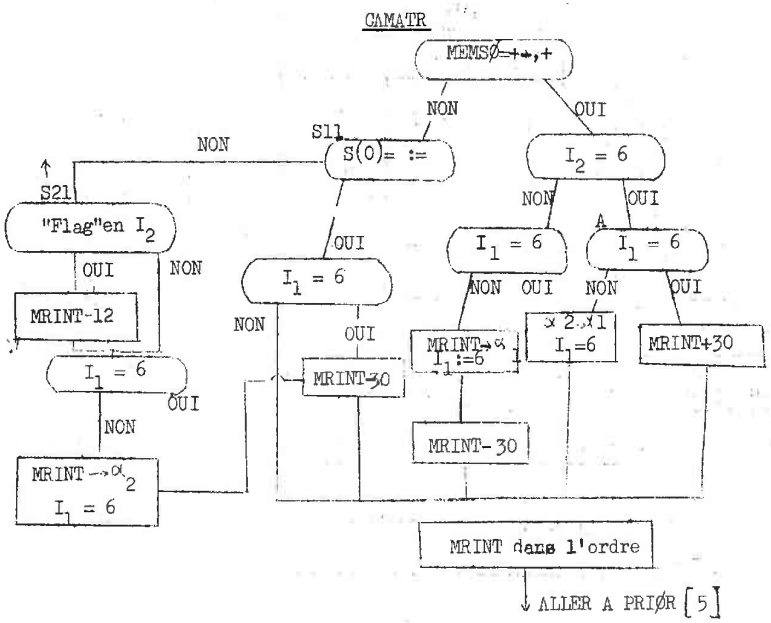
si $I_1 = I_2 = \bar{5}$

si I_1 ou $I_2 = 6$

si $I_1 = I_2 = \bar{5}$

si I_1 ou $I_2 = 6$

ORDINOGRAMME 3



ANNEXE

NOTA : Les signes + du programme sont représentés par &
La paranthèse ouverte est représentée par %
La paranthèse fermée est représentée par ¤

```
*SOUS PROGRAMME DE CONSTRUCTION DE VECTEUR DE RENSEIGNEMENT
BUTOIRDS   ,1752
INAC  DS   ,3081
TRAD  DS   ,4538
BB    DS   ,4707
      DS   10
DEBUT S    *-1,INAC,11
      TFM  TRAD-1,DEBUT-6
      TDM  BB,9
      TFM  BB&5,*&20
      B    TRAD
      DORG*-4
      TDM  BB,2
      TFM  N,0,10
      TFM  RO,0,7
      TFM  T3,0,7
      SM   DEBUT-1,5,10
      MM   DEBUT-1,7,610
      S    DEBUT-1,99
      TF   T3,99
      SM   DEBUT-1,2,10
      TD   T1,DEBUT-1,11
      CF   T1
CN    DS   ,*
      SM   DEBUT-1,2,10
      TF   T2,DEBUT-1,11
      SM   DEBUT-1,1,10
      TF   T5,DEBUT-6,
DEPARTTFM  CN,0,7
      SM   DEBUT-6,4,10
      SF   DEBUT-6,,6
      AM   DEBUT-6,4,10
      S    DEBUT-6,DEBUT-1,611
      BN   E1
      SM   DEBUT-1,5,10
      C    DEBUT-1,DEBUT-6,611
      BH   *&24
      B    E2
      M    RO,DEBUT-1,11
      SF   95
T1    DC   2,0,*
T2    DC   2,0,*-2
      TF   RO,99
      A    RO,DEBUT-6,11
      AM   N,1,10
      SM   DEBUT-1,5,10
      C    N,T2
      BE   *&36
      SM   DEBUT-6,12,10
      B    DEPART
COMPT DS   ,*
N     DS   ,*-2
      MM   N,10,10
      A    T3,99
      AM   T3,9,10
      A    DEBUT-1,T3
      CM   T1,5,10
```

```
        BNE *&48
        MM  RO,12,10
        SF  95
T3      DS  ,*
        TF  RO,99
        TF  T5,DEBUT-1,611
        A   T5,RO,6
        BB
        DORG*-10
E1      BTM BUTOIR,11,9
E2      BTM BUTOIR,12,9
E3      BTM BUTOIR,13,9
E4      BTM BUTOIR,14,9
E5      BTM BUTOIR,15,9
```



```
*SOUS PROGRAMME  DE  CACUL  D ADRESSE,
        DORG*&31
DEB3    S   *-1,INAC,11
        TFM TRAD-1,DEB3-14
        TDM BB,9
        TFM BB&5,*&20
        B   TRAD
        DORG*-4
        TDM BB,2
        TFM COMPT,0,10
        TFM TRAV,1,7
        TFM N,0,10
        TD  N,DEB3-13,
        CF  N
T5      DS  ,*
        TF  T1,DEB3-6
        TF  T2,DEB3-19
        CF  T2
RO      DS  ,*
        SM  DEB3-14,5,10
        TFM DEB3-14,0,610
        SM  DEB3-14,2,10
        TD  DEB3-14,DEB3-13,6
        SM  DEB3-14,2,10
        TF  DEB3-14,T1,6
        SM  DEB3-14,1,10
ET      SM  DEB3-1,4,10
        SF  DEB3-1,,6
        SM  DEB3-1,1,10
        CM  DEB3-1,0,67
        BNE E3
        AM  DEB3-1,5,10
        TF  DEB3-14,DEB3-1,611
        SM  DEB3-1,12,10
        SM  DEB3-1,4,10
        SF  DEB3-1,,6
        AM  DEB3-1,4,10
        S   DEB3-1,DEB3-14,611
        AM  DEB3-1,1,610
        BN  E5
        SM  DEB3-1,5,10
        CM  DEB3-1,0,67
        BNE E4
        AM  DEB3-1,5,10
        SM  DEB3-14,5,10
        TF  DEB3-14,DEB3-1,611
        M   DEB3-14,TRAV,6
        SF  95
TRAV    DS  *,5
        TF  TRAV,99
        AM  COMPT,1,10
        SM  DEB3-14,5,10
        C   COMPT,T1
        BF  *&36
        SM  DEB3-1,12,10
        B   ET
        CM  N,5,10
```

```
      BNE *&48
      MM  TRAV,12,10
      SF  95
      TF  TRAV,99
      MM  T1,10,10
      AM  99,10,10
      A   DEB3-14,99
      TF  DEB3-14,DEB3-8,611
      S   DEB3-14,TRAV,6
      TFM COMPT,0,10
TEST10C   COMPT,T2
      BE  FINBAB
      AM  COMPT,1,10
      AM  DEB3-14,2,10
      TF  DEB3-14,COMPT,6
      AM  DEB3-14,5,10
      TF  DEB3-14,DEB3-8,611
      AM  COMPT,1,10
      M   TRAV,COMPT,
      SF  95
      S   DEB3-14,99,6
      SM  COMPT,1,10
      B   TEST10
FINBABTF  CN,DEB3-14,11
      BB
      DORG*-10
```



```
*COMPILATION DES TABLEAUX LOCAUX ET REMANENTS,
MI    DC  2,0,
T3    DC  2,0,
T5    DS  5,
CTAB1 DS  5,
T2    DC  2,0,
T1    DC  2,0
N     DC  2,0,
Z2    DSC 13,260000000000@,
Z4    TFM BAB2-14
      TDM BAB2-13
      TFM BAB2-8
      TFM BAB2-19
      TFM BAB2-6
      BTM BAB2
      DC  1,@,
Z3    TFM BAB1-6
      BTM BAB1
      DC  1,@
BAB2  DS  ,12722
BAB1  DS  ,12010
INI2  DC  5,49994
INI   DC  5,49999
CTAB  DS  5,
CTAB2 DS  5,
NP    DC  2,0,
TABLO DS  5,
CROCH SM  MEMSV,5,10
      CF  SYMBOL
PVIRGUSM  MEMSV,1,10
      BNF *&24,FL2T
      AM  MRINT,5,10
      BNF *&108,FLR2
      TFM C4&29,24,10
      TR  C4,ZAFF1
      TF  C4&11,MEMSV,11
      BNF *&36,FL2RP
      TF  C4&18,MEMSV,11
      B   *&120
      TF  C4&18,MRINT
      B   *&84
      BNF *&24,FL2RP
      B   *&96
      TFM C4&29,12,10
      TR  C4,Z2
      TF  C4&6,MRINT
      TF  C4&11,MEMSV,11
      SM  MRINT,12,10
      TR  MPGOB,C4,6
      A   MPGOB,C4&29
      AM  MEMSV,7,10
      TD  T1,MEMSV,11
      CF  T1
      CM  T1,5,10
      BE  *&36
      CM  T1,6,10
      BNE *&72
      SM  MEMSV,7,10
```

```
        AM   MRINT,12,10
        TF   MEMSV,MRINT,6
        SM   MRINT,12,10
        AM   MEMSV,1,10
        SM   MEMSO,2,10
        CM   MEMSO,40,610
        BE   *&24
        AM   N,1,10
        AM   MEMSO,2,10
        CM   MEMSE,44,610
        BNE  *&60
        SM   MEMSO,2,10
        CM   MEMSO,40,610
        BE   JCROCH
        B    ZCROCH
        CM   MEMSE,34,610
        BE   VRBCL
        B    BOUCLE
DEPOINSM  MEMSV,5,10
        SM   MEMSO,2,10
        AM   N,1,10
        B    PRIOR
TBLEA   CM   MEMSE,24,610
        BE   *&132
        BD   *&36,NP
        BNF  *&24,NP
        B    *&72
        AM   MEMSV,2,610
        SF   MEMSV
        TFM  MEMSV,0,69
        CF   MEMSV
        SM   MEMSV,2,610
        AM   NP,1,10
        B    BOUCLE
        SM   MRINT,24,10
        MM   NP,7,10
        S    MRINT,99
        SF   TBLEA
        SM   MEMSO,2,10
        CM   MEMSO,31,610
        BE   TNR
        CM   MEMSO,39,610
        BE   *&84
        SM   MEMSO,2,10
        CM   MEMSO,39,610
        BNE  TNR-12
        TF   T2,MEMSO,11
        AM   MEMSO,2,10
        B    *&24
        TF   T2,MEMSO,11
        AM   MEMSO,2,10
        SM   MEMSV,7,10
        TFM  MEMSV,99999,67
        AM   MEMSV,1,10
        TDM  MEMSV,5,6
        B    VRBCL
        AM   MEMSO,2,10
TNR     AM   MEMSO,2,10
```



```
        BT   PORTE,CPTR2
        SM   MEMSV,7,10
        TF   MEMSV,PORTE-1,6
        AM   MEMSV,1,10
        TDM  MEMSV,5,6
        B    VRBCL
        DORG*&6
PORTE   TF   T5,PORTE-1,
        SM   PORTE-1,24,10
        SM   T5,41,10
        CM   PORTE-1,C13-2,7
        BE   P10
        BNF  P12,T5,11
        AM   T5,2,10
        C    CTNIV,T5,11
        BNE  *&72
        AM   T5,4,10
        CF   T5,,6
        TF   PORTE-1,TABLO
        SF   PORTE-1,,
        BB
        SM   T5,2,10
P12     TD   T1,PORTE-1,11
        CF   T1
        CM   T1,5,10
        BE   *&72
        CM   T1,6,10
        BF   *&48
        SM   PORTE-1,22,10
        SM   T5,22,10
        B    PORTE&36
        SM   PORTE-1,1,10
        TF   PORTE-1,PORTE-1,11
        BB
P10     TF   PORTE-1,INI
        BB
        DORG*-10
JCROCHSM  MEMSV,1,10
        CM   T2,39,10
        BNE  SUITE
        MM   NP,7,10
        TF   T5,99
        SM   T5,7,10
        S    REMAN,T5
        TR   C4,ZRELA&44
        TF   C4&6,REMAN
        AM   C4&6,1,10
        A    REMAN,T5
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        TR   C4,TFM1,
        TF   T5,MPGOB
        SM   T5,C3
        A    T5,CHOB
        TF   C4&6,T5
        SM   C4&6,10,10
        TFM  C4&11,41,10
        TR   MPGOB,C4,6
```

```
        AM   MPGOB,12,10
SUITE   TR   C4,Z4
        TF   C4&71,MEMSV,11
        TF   MRINT,MEMSV,11
        AM   MRINT,24,10
        MM   NP,7,10
        A    MRINT,99
        TF   C4&47,NP
        TF   C4&59,N
        AM   MEMSV,6,10
        TF   C4&35,MEMSV,11
        SM   MEMSO,2,10
        AM   MEMSV,1,10
        CM   T2,39,10
        BNE  *&36
        TF   CTAB,REMAN
        B    *&24
        TF   CTAB,MRINT
JO      AM   MEMSV,6,10
        SF   MEMSV
        CM   MEMSO,37,610
        BE   *&48
        CM   MEMSO,38,610
        BE   *&48
        B    *&60
        TDM  MEMSV,5,6
        B    *&48
        TDM  MEMSV,6,611
        B    *&24
        TDM  MEMSV,5,611
        CF   MEMSV
        SM   MEMSV,1,610
        SF   MEMSV
        TF   MEMSV,CTAB,6
        CF   MEMSV
        CM   T2,39,10
        BNE  *&72
        AM   MEMSV,2,610
        SF   MEMSV
        TDM  MEMSV,9,6
        CF   MEMSV
        SM   MEMSV,2,610
        AM   MEMSV,3,610
        SF   MEMSV
        TD   MEMSV,N,6
        CF   MEMSV
        SM   MEMSV,3,610
        CM   NP,0,10
        BE   *&48
        SM   CTAB,7,10
        SM   NP,1,10
        B    JO
        MM   N,10,10
        AM   99,10,10
        S    CTAB,99
        CM   T2,39,10
        BNE  *&36
        TF   REMAN,CTAB
```



```
        B    *&24
        TF   MRINT,CTAB
        SF   MEMSV
        TF   C4&11,MEMSV,11
        CF   MEMSV
        AM   MEMSV,1,610
        SF   MEMSV
        TD   C4&23,MEMSV,11
        CF   MEMSV
        TR   MPGOB,C4,6
        AM   MPGOB,72,10
        TFM  NP,1,1011
        TFM  N,0,10
        AM   MEMSO,2,10
        AM   MEMSV,6,10
        CF   TBLEA
        B    SYMBOL
ZCROCHTR C4,Z3
        SM   MEMSV,1,10
        TF   C4&11,MEMSV,11
        TF   MRINT,MEMSV,11
        AM   MEMSV,5,10
        CF   MEMSV,,6
        AM   MEMSV,1,10
        TF   C4&23,MEMSV,11
        AM   MEMSV,1,10
        TD   T1,MEMSV,11
        CF   T1
        TD   T5,MEMSV,11
        AM   MEMSV,6,10
        BNF  *&36,MEMSV,11
        TD   MEMSV,N,6
        B    *&72
        SF   MEMSV
        TD   MI,MEMSV,11
        C    MI,N,
        BE   *&24
        BTM  IMPER,4,9
        CF   MEMSV
        TFM  N,0,10
        SM   MEMSV,1,10
        TF   MEMSV,MRINT,6
        SF   MEMSV,,6
        SM   MRINT,5,10
        AM   MEMSV,1,10
        CM   T1,5,10
        BE   *&36
        TDM  MEMSV,3,6
        B    *&60
        BNF  *&36,T5
        TDM  MEMSV,0,6
        B    *&24
        TDM  MEMSV,1,6
        TR   MPGOB,C4,6
        AM   MPGOB,24,10
        B    SYMBOL
TAB2    SM   MEMSV,5,10
        TFM  NP,0,10
```

```
SM   MEMSO,2,10
CM   MEMSO,37,610
BE   *&60
CM   MEMSO,36,610
BE   *&36
CM   MEMSO,38,610
BNE  *&24
SM   MEMSO,2,10
CM   T2,39,10
BNE  PRIOR
SM   MEMSO,2,10
TFM  T2,0,10
B    PRIOR
```



```
*EXECUTION DES TABLEAUX REMANENTS
*ASSEMBLER SEPAREMENT EN DEFINISSANT BB TRAD INAC BUTOIR
       DS   10
DEBUT  S    *-1,INAC,11
       TFM  TRAD-1,DEBUT-6
       TDM  BB,9
       TFM  BB&5,*&20
       B    TRAD
       DORG*-4
       TDM  BB,2
       TFM  N,0,10
       TFM  R0,0,7
       TFM  T3,0,7
       SM   DEBUT-1,5,10
       MM   DEBUT-1,7,610
       S    DEBUT-1,99
       TF   T3,99
       SM   DEBUT-1,2,10
       TD   T1,DEBUT-1,11
       CF   T1
CN     DS   ,*
       SM   DEBUT-1,2,10
       TF   T2,DEBUT-1,11
       SM   DEBUT-1,1,10
       TF   T5,DEBUT-6,
DEPARTTFM   CN,0,7
       SM   DEBUT-6,4,10
       SF   DEBUT-6,,6
       AM   DEBUT-6,4,10
       S    DEBUT-6,DEBUT-1,611
       BN   E1
       SM   DEBUT-1,5,10
       A    CN,DEBUT-1,11
       SM   CN,1,10
       S    CN,DEBUT-6,11
       BN   E2
       M    R0,DEBUT-1,11
       SF   95
T1     DC   2,0,*
T2     DC   2,0,*-2
       TF   R0,99
       A    R0,DEBUT-6,11
       AM   N,1,10
       SM   DEBUT-1,5,10
       C    N,T2
       BE   *&36
       SM   DEBUT-6,12,10
       B    DEPART
COMPT  DS   ,*
N      DS   ,*-2
       MM   N,10,10
       A    T3,99
       AM   T3,10,10
       A    DEBUT-1,T3
       CM   T1,5,10
       BNE  *&48
       MM   R0,12,10
       SF   95
```

```
T3      DS   ,*
        TF   R0,99
        TF   T5,DEBUT-1,611
        A    T5,R0,6
        BB
        DORG*-10
E1      BTM  BUTOIR,12,9
E2      BTM  BUTOIR,12,9
E3      BTM  BUTOIR,13,9
E4      BTM  BUTOIR,14,9
E5      BTM  BUTOIR,15,9
K2      DS   2,
K1      DS   2,
        DORG*&31
DEB3    S    *-1,INAC,11
        CF   DEB3
        CF   DEBUT
        TFM  TRAD-1,DEB3-14
        TDM  BB,9
        TFM  BB&5,*&20
        B    TRAD
        DORG*-4
        TDM  BB,2
        TFM  COMPT,0,10
        TFM  TRAV,1,7
        TFM  N,0,10
        TD   N,DEB3-13,
        BNF  *&24,N
        SF   DEBUT
        CM   N,5,10
        BE   *&48
        TFM  K1,1,10
        TFM  K2,1,10
        B    *&36
        TFM  K1,12,10
        TFM  K2,11,10
        CF   N
T5      DS   ,*
        TF   T1,DEB3-6,
        TF   T2,DEB3-19
        CF   T2
R0      DS   ,*
        SM   DEB3-14,5,10
        TFM  DEB3-14,0,610
        SM   DEB3-14,2,10
        TD   DEB3-14,DEB3-13,6
        SM   DEB3-14,2,10
        TF   DEB3-14,T1,6
        SM   DEB3-14,1,10
        CM   DEB3-6,99999,7
        BE   TR1
ET      SM   DEB3-1,4,10
        SF   DEB3-1,,6
        SM   DEB3-1,1,10
        CM   DEB3-1,0,67
        BNF  E3
        AM   DEB3-1,5,10
        SM   DEB3-1,4,10
```



```
        TF   DEB3-14,DEB3-1,611
        SM   DEB3-1,12,10
        SF   DEB3-1,,6
        AM   DEB3-1,4,10
        S    DEB3-1,DEB3-14,611
        AM   DEB3-1,1,610
        BN   E5
        SM   DEB3-1,5,10
        CM   DEB3-1,0,67
        BNE  E4
        AM   DEB3-1,5,10
        SM   DEB3-14,5,10
        TF   DEB3-14,DEB3-1,611
        M    DEB3-14,TRAV,6
        SF   95
        TF   TRAV,99
        AM   COMPT,1,10
        SM   DEB3-14,5,10
        C    COMPT,T1
        BE   *&36
        SM   DEB3-1,12,10
        B    ET
        M    TRAV,K1
        SF   95
        TF   TRAV,99
        CM   DEB3-8,99999,7
        BE   TR2
        MM   T1,10,10
        AM   99,10,10
        A    DEB3-14,99
        TF   DEB3-14,DEB3-8,611
        S    DEB3-14,TRAV,6
        TFM  COMPT,0,10
TEST10C CM   COMPT,T2
        BE   FINBAB
        AM   COMPT,1,10
        AM   DEB3-14,2,10
        TF   DEB3-14,COMPT,6
        AM   DEB3-14,5,10
        TF   DEB3-14,DEB3-8,611
        AM   COMPT,1,10
        M    TRAV,COMPT,
        SF   95
        S    DEB3-14,99,6
        SM   COMPT,1,10
        B    TEST10
FINBABTF CN,AMINI
        BB
        DORG*-10
TR1     TF   AMINI1,DEB3-14
        SM   DEB3-14,7,10
        BD   *&48,DEB3-14,11
        SF   DEB3
        AM   DEB3-14,7,10
        B    ET
        TF   DEB3-14,DEB3-1
        B    ET
TRAV    DS   ,*
```

```
TR2    BNF *&24,DEB3
       B   INITIA
       S   AMINI,TRAV
       AM  AMINI1,10,10
       TF  AMINI1,AMINI,6
       TF  NOM,AMINI1,6
       AM  NOM,5,10
       AM  COMPT4,1,10
       TFM COMPT,0,10
TEST3  C   COMPT,T2
       BE  FINBAB
       AM  COMPT,1,10
       AM  AMINI1,2,10
       TF  AMINI1,COMPT,6
       AM  AMINI1,5,10
       S   AMINI,TRAV
       TF  AMINI1,AMINI,6
       TF  NOM,AMINI1,6
       AM  NOM,5,10
       AM  COMPT4,1,10
       B   TEST3
AMINI  DS  ,49994
NOM    DS  ,49989
COMPT4DS  ,49984
AMINI1DS  5,
AMINI2DS  5
AMINI3DS  5,
AMINI4DS  5,
AMINI5DS  5,
A1     DS  5,
A2     DS  5,
C1     DS  5,
C2     DS  5,
C3     DS  5,
C4     DS  5,
REMARKDC  2,@,
TRANSFDS  5,
RANSF1DS  5,
AMAX   DS  5,
M      DS  5,
DIGIT  DS  2,
COMPT2DC  2,0,
COMPT3DC  2,0,
P      DS  5,
TRAV1  DS  5,
INITIAMM  T1,10,10
       A   DEB3-14,99
       AM  AMINI1,10,10
       TF  AMINI2,AMINI1,11
       TF  AMINI4,AMINI1
       SM  AMINI1,10,10
       TFM TRAV1,1,7
       TFM COMPT,1,10
       TF  TRANSF,AMINI1
X      SM  AMINI1,5,10
       M   TRAV1,AMINI1,11
       SF  95
       TF  TRAV1,99
```



```
       C   COMPT,T1
       BE  *&48
       SM  AMINI1,5,10
       AM  COMPT,1,10
       B   X
       M   TRAV,K1
       SF  95
       TF  TRAV1,99
       TFM COMPT3,0,10
       TF  AMINI1,TRANSF,
NF     TF  P,AMINI2,
       S   P,AMINI,
       TF  AMINI5,AMINI
       TFM COMPT2,0,10
       TFM C1,K1
       TFM C2,K1
       TFM A1,K1
       TFM A2,K1
       MM  T1,10,10
       S   AMINI1,99
       S   DEB3-14,99
       AM  AMINI1,10,10
       AM  DEB3-14,10,10
COMP1  M   A1,AMINI1,11
       SF  95
       TF  A1,99
       SM  AMINI1,5,10
       M   C1,AMINI1,11
       SF  95
       TF  C1,99
       M   A2,DEB3-14,11
       SF  95
       TF  A2,99
       SM  DEB3-14,5,10
       M   C2,DEB3-14,11
       SF  95
       TF  C2,99,
       AM  AMINI1,15,10
       AM  DEB3-14,15,10
       AM  COMPT2,1,10
       C   A1,A2
       BNE ESPACE
       C   C1,C2
       BNE ESPACE
       C   COMPT2,T1
       BNE TR5
       BD  E,DIGIT-1
       BB
TR5    TF  A1,C2
       TF  A2,C2
       TF  C1,C2
       B   COMP1
ESPACETDM DIGIT-1,9
       TF  COMPT,COMPT2
       TF  TRANSF,AMINI1
       TF  C3,C2
       TF  C4,C1
A      C   COMPT,T1
```

```
        BE   B
        SM   TRANSF,5,10
        M    C3,TRANSF,11
        SF   95
        TF   C3,99
        M    C4,TRANSF,11
        SF   95
        TF   C4,99
        AM   TRANSF,15,10
        AM   COMPT,1,10
        B    A
B       CM   P,0,7
        BE   C
        C    C2,C1
        BNH  C&24
        TF   TRANSF,AMINI2
        S    TRANSF,C3
        A    TRANSF,C4,
        S    TRANSF,P
        TD   DIGIT,AMINI2,11
        TD   AMINI2,REMARK,6
        TF   RANSF1,AMINI2
        S    RANSF1,P
        S    TRANSF,K2,10
        S    RANSF1,K2,10
        TR   TRANSF,RANSF1,611
        TD   AMINI2,DIGIT,6
        A    TRANSF,P,
        A    TRANSF,K2
        TD   TRANSF,DIGIT,6
D       S    AMINI,C3
        A    AMINI,C4
        TF   AMINI3,AMINI2
        S    AMINI3,C3
        A    AMINI3,C4
        B    MOVE
C       C    C2,C1
        BH   D
        TF   AMINI3,AMINI2
MOVE    TF   AMAX,AMINI2
        A    AMAX,C4
        TF   M,A2
        S    M,A1
        TF   TRANSF,C1
        C    M,TRANSF
        BNH  TR5
        TFM  TRANSF,1,10
        S    TRANSF,C2
        C    M,TRANSF,
        BNH  TR5
TEST2   C    A1,A2
        BNH  TRANS1
        C    C2,C1
        BH   *&24
        B    *&60
        TF   TRANSF,AMINI3
        S    TRANSF,AMINI2
        C    TRANSF,M
```



```
        BNH  TRANS1
        TFM  XYZ&1,26,10
        TF   XYZ&6,AMINI3
        TF   XYZ&11,AMINI2
        CM   N,6,10
        BNE  *&36
        TFM  XYZ&1,25,10
        B    *&36
        BNF  *&24,DEBUT
        TDM  XYZ,0
        A    XYZ&6,C2
        A    XYZ&11,C2
        A    XYZ&11,M
XYZ     TF   ,,611
        S    XYZ&6,K1
        S    XYZ&11,K1
        C    AMINI2,XYZ&11
        BNE  XYZ
        A    AMINI2,C1
        A    AMINI3,C2
        B    VIC
TRANS1A AMINI2,C1
        TD   DIGIT,AMINI2,11
        TD   AMINI2,REMARK,6,
        TF   TRANSF,C1
        S    TRANSF,M
        TF   XY&11,AMINI2
        TF   XY&6,AMINI3
        A    XY&11,M
        S    AMINI2,K2
        S    AMINI3,K2
XY      TR   ,,611
        A    AMINI3,TRANSF
        TD   AMINI3,DIGIT,6
        S    AMINI3,TRANSF
        A    AMINI3,C2
        A    AMINI2,C1
        TD   AMINI2,DIGIT,6
        A    AMINI2,K2
        A    AMINI3,K2
VIC     C    AMAX,AMINI2
        BNE  TEST2
        S    AMINI3,C3
        TF   AMINI2,AMINI3
        C    COMPT2,T1
        BNE  TR5
E       SM   DEB3-14,10,10
        SM   AMINI1,10,10
TR4     CM   P,0,7
        BNE  PUIS
        A    AMINI4,TRAV1,6
        S    AMINI4,TRAV,6
        B    OK1
PUIS    TFM  COMPT,0,10
        TF   TRANSF,49989
        MM   49984,5,10
        S    TRANSF,99
        C    TRANSF,AMINI4,6
```

```
        BE   *&48
        AM   COMPT,1,10
        AM   TRANSF,5,10
        B    *-48
        MM   COMPT,5,10
        A    99,49989,
        TF   TRANSF,99,7
        MM   49984,5,10
        S    TRANSF,99
OK      AM   COMPT,1,10
        TF   RANSF1,TRANSF,11
        A    RANSF1,TRAV1,6
        S    RANSF1,TRAV,6
        AM   TRANSF,5,10
        C    COMPT,COMPT4
        BNE  OK
OK1     TF   TRANSF,AMINI5
        S    TRANSF,AMINI
        TF   RANSF1,TRAV1
        S    RANSF1,TRAV
        CF   RANSF1
        C    RANSF1,TRANSF,
        BE   TR6
        A    TRANSF,RANSF1
        TF   RANSF1,AMINI
        A    RANSF1,P
        A    RANSF1,TRAV
        TF   T5,RANSF1
        A    T5,TRANSF
        A    AMINI,TRANSF
        TFM  Z&1,26,10
        CM   N,6,10
        BNE  *&36
        TFM  Z&1,25,10
        B    *&36
        BNF  *&24,DEBUT
        TDM  Z,0
Z       TF   T5,TRANSF,611
        S    Z&6,K1
        S    Z&11,K1
        C    Z&6,AMINI
        BNE  Z
TR6     C    COMPT3,T2
        BE   *&60
        AM   COMPT3,1,10
        AM   AMINI4,7,10
        TF   AMINI2,AMINI4,11
        B    NF
        MM   T1,10,10
        TD   DIGIT,DEB3-14,11
        TD   DEB3-14,REMARK,6
        S    DEB3-14,99
        S    AMINI1,99
        TR   AMINI1,DEB3-14,611
        TDM  DIGIT-1,0,
        A    AMINI1,99
        TD   AMINI1,DIGIT,6
        BB
        DORG*-10
        DEND20000
```



```
*COMPILATION DES TABLEAUX LOCAUX
CTAB1   DS   5,
T1      DC   2,0
N       DC   2,0,
Z4      TFM  BAB2-14
        TDM  BAB2-13
        TFM  BAB2-8
        TFM  BAB2-19,
        TFM  BAB2-6
        BTM  BAB2
        DC   1,@,
Z2      DSC  13,2600000000000@,
Z3      TFM  BAB1-6
        BTM  BAB1
        DC   1,@,
BAB1    DS   ,5360
BAB2    DS   ,6072
INI     DC   5,49999
T5      DS   5,
CTAB2   DS   5,
NP      DC   2,0,
CROCH   SM   MEMSV,5,10
        CF   SYMBOL
        B    PVIRGU
TBLEA   CM   MEMSE,24,610
        BE   *&108
        BD   *&36,NP
        BNF  *&24,NP
        B    *&48
        SF   MEMSV
        TDM  MEMSV,0,6
        CF   MEMSV
        AM   NP,1,10
        B    BOUCLE
        SM   MRINT,24,10
        MM   NP,7,10
        S    MRINT,99,
        SF   TBLEA
        BT   PORTE,CPTR2
        SM   MEMSV,7,10
        TF   MEMSV,PORTE-1,6
        AM   MEMSV,1,10
        TDM  MEMSV,5,6
        B    VRBCL
        DORG*&6
PORTE   TF   T5,PORTE-1,
        SM   PORTE-1,24,10
        SM   T5,41,10
        CM   PORTE-1,C13-2,7
        BE   P10
        BNF  P12,T5,11
        AM   T5,2,10
        C    CTNIV,T5,11
        BNE  *&72
        AM   T5,4,10
        CF   T5,,6
        TF   PORTE-1,TABLO
        SF   PORTE-1
        BR
        SM   T5,2,10
```

```
P12     TD   T1,PORTE-1,11
        CF   T1
        CM   T1,5,10
        BE   *&72
        CM   T1,6,10
        BE   *&48
        SM   PORTE-1,22,10
        SM   T5,22,10
        B    PORTE&36
        SM   PORTE-1,1,10
        TF   PORTE-1,PORTE-1,11
        BB
P10     TF   PORTE-1,INI
        BB
        DORG*-10
PVIRGUSM  MEMSV,1,10
        BNF  *&24,FL2T
        AM   MRINT,5,10
        BNF  *&108,FLR2
        TFM  C4&29,24,10
        TR   C4,ZAFF1
        TF   C4&11,MEMSV,11
        BNF  *&36,FL2RP
        TF   C4&18,MEMSV,11
        B    *&120
        TF   C4&18,MRINT
        B    *&84
        BNF  *&24,FL2RP
        B    *&96
        TFM  C4&29,12,10
        TR   C4,Z2
        TF   C4&6,MRINT
        TF   C4&11,MEMSV,11
        SM   MRINT,12,10
        TR   MPGOB,C4,6
        A    MPGOB,C4&29
        AM   MEMSV,7,10
        TD   T1,MEMSV,11
        CF   T1
        CM   T1,5,10
        BE   *&36
        CM   T1,6,10
        BNE  *&72
        SM   MEMSV,7,10
        AM   MRINT,12,10
        TF   MEMSV,MRINT,6
        SM   MRINT,12,10
        AM   MEMSV,1,10
        SM   MEMSO,2,10
        CM   MEMSO,40,610
        BE   *&24
        AM   N,1,10
        AM   MEMSO,2,10
        CM   MEMSE,44,610
        BNE  *&60
        SM   MEMSO,2,10
        CM   MEMSO,40,610
        BE   JCROCH
        B    ZCROCH
        CM   MEMSE,34,610
```



```
        BE   VRBCL
        B    BOUCLE
DEPOINSM  MEMSV,5,10
        SM   MEMSO,2,10
        AM   N,1,10
        B    PRIOR
JCROCHSM  MEMSV,1,10
        TR   C4,Z4
        TF   C4&71,MEMSV,11
        TF   MRINT,MEMSV,11
        AM   MRINT,24,10
        MM   NP,7,10
        A    MRINT,99
        TF   C4&47,NP
        TF   C4&59,N
        AM   MEMSV,6,10
        TF   C4&35,MEMSV,11
        SM   MEMSO,2,10
        AM   MEMSV,1,10
        AM   MEMSV,6,10
        SF   MEMSV
        CM   MEMSO,37,610
        BE   *&48
        CM   MEMSO,38,610
        BE   *&48
        B    *&60
        TDM  MEMSV,5,6
        B    *&48
        TDM  MEMSV,6,611
        B    *&24
        TDM  MEMSV,5,611
        CF   MEMSV
        SM   MEMSV,1,610
        SF   MEMSV
        TF   MEMSV,MRINT,6
        CF   MEMSV
        AM   MEMSV,3,610
        SF   MEMSV
        TF   MEMSV,N,6
        CF   MEMSV
        SM   MEMSV,3,610
        CM   NP,0,10
        BE   *&48
        SM   NP,1,10
        SM   MRINT,7,10
        B    *-312
        MM   N,10,10
        AM   99,10,10
        S    MRINT,99
        SF   MEMSV
        TF   C4&11,MEMSV,11
        CF   MEMSV
        AM   MEMSV,1,610
        SF   MEMSV
        TD   C4&23,MEMSV,11
        CF   MEMSV
        TR   MPGOB,C4,6
        AM   MPGOB,72,10
        TFM  NP,1,1011
        TFM  N,0,10
```

```
        AM   MEMSO,2,10
        AM   MEMSV,6,10
        CF   TBLEA
        B    SYMBOL
ZCROCHTR C4,Z3
        SM   MEMSV,1,10
        TF   C4&11,MEMSV,11
        TF   MRINT,MEMSV,11
        AM   MEMSV,5,10
        CF   MEMSV,,6
        AM   MEMSV,1,10
        TF   C4&23,MEMSV,11
        AM   MEMSV,1,10
        TD   T1,MEMSV,11
        CF   T1
        TD   T5,MEMSV,11
        AM   MEMSV,6,10
        SF   MEMSV,,,
        BNF  *&36,MEMSV,11
        TF   MEMSV,N,6
        B    *&48
        C    N,MEMSV,11
        BE   *&24
        BTM  IMPER,4,9
        CF   MEMSV
        TFM  N,0,10
        SM   MEMSV,1,10
        TF   MEMSV,MRINT,6
        SF   MEMSV,,6
        SM   MRINT,5,10
        AM   MEMSV,1,10
        CM   T1,5,10
        BE   *&36
        TDM  MEMSV,3,6
        B    *&60
        BNF  *&36,T5
        TDM  MEMSV,0,6
        B    *&24
        TDM  MEMSV,1,6
        TR   MPGOB,C4,6
        AM   MPGOB,24,10
        B    SYMBOL
TAB2    SM   MEMSV,5,10
        TFM  NP,0,10
        SM   MEMSO,2,10
        CM   MEMSO,37,610
        BE   *&60
        CM   MEMSO,38,610
        BE   *&36
        CM   MEMSO,36,610
        BNE  PRIOR
        SM   MEMSO,2,10
        B    PRIOR
TABLO   DC   5,3101
```



```
*COMPILATION D ALGOL MATRICIEL,
        DORG47500
BAB3    DS   ,30000
BEGIN   DS   ,6182
MEMSV   DS   ,416
T1      DS   ,38890
PORTE   DS   ,39304
CPTR2   DS   ,2471
C4      DS   ,2154
MEMSO   DS   ,411
MRINT   DS   ,431
T5      DS   ,39014
MPGOB   DS   ,426
PRIOR   DS   ,14032
FL2RP   DS   ,2439
MEMSE   DS   ,406
GAMSP   DS   ,23910
GAMMA   DS   ,14908
T2      DC   2,0,
T3      DC   2,0,
Z8      TFM  BAB3⌙20,
        DC   1,@,
Z9      TDM  BAB3⌙25,
        DC   1,@,
Z7      TFM  BAB3⌙18,
        TFM  BAB3⌙13
        TFM  BAB3⌙8
        TDM  BAB3⌙7
        TDM  BAB3⌙6
        BTM  BAB3
        DC   1,@
CAMATRTD T2,MEMSV,11
        AM   MEMSV,12,10
        CF   T2
        CM   T1,6,10
        BE   S31
        CM   T2,6,10
        BE   S31
        TF   T5,T3
        TD   T3,MEMSV,11
        CF   T3
        CM   T3,6,10
        BE   S31
        BT   PORTE,CPTR2
        TR   C4,Z8
        TF   C4&11,PORTE⌙1
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        TR   C4,Z9
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        B    *&72
S31     TR   C4,Z9
        TDM  C4&11,9
        TR   MPGOB,C4,6
        AM   MPGOB,12,10
        SM   MEMSV,6,10
        TR   C4,Z7
        TD   C4&59,MEMSV,11
        SM   MEMSV,1,10
```

```
        TF   C4&71,MEMSV,11
        SM   MEMSV,5,10
        TD   C4&47,MEMSV,11
        SM   MEMSV,1,10
        TF   C4&35,MEMSV,11
        AM   MEMSV,7,10
        TF   C4&11,MEMSO,11
        CM   MEMSO,8,610
        BE   S21
        CM   MEMSO,25,610
        BE   S11
        CM   T2,6,10
        BE   A
        CM   T1,6,10
        BE   END
        TDM  MEMSV,6,6
        SM   MEMSV,1,10
        TF   MEMSV,MRINT,6
        AM   MEMSV,1,10
        SM   MRINT,30,10
        B    END
A       CM   T1,6,10
        BNE  *&36
        AM   MRINT,30,10
        B    END
        SM   MEMSV,7,10
        TF   T5,MEMSV,11
        AM   MEMSV,6,10
        TF   MEMSV,T5,6
        AM   MEMSV,1,10
        TDM  MEMSV,6,6
END     TF   C4&23,MRINT
        SM   MEMSO,2,10
        TR   MPGOB,C4,6
        AM   MPGOB,72,10
        B    PRIOR
S11     AM   MEMSV,6,10
        CM   T1,6,10
        BNE  END
        AM   MRINT,30,10
        B    END
S21     BNF  *&24,FL2RP
        AM   MRINT,12,10
        CM   T1,6,10
        BE   END
        TDM  MEMSV,6,6
        SM   MEMSV,1,10
        TF   MEMSV,MRINT,6
        AM   MRINT,30,10
        AM   MEMSV,1,10
        B    END
```



```
*DEBUT DE GAMMA DE%5¤ POUR L ALGOL MATICIEL
GAMMA1CM   MEMSE,24,610
        BE   GAMSP
        AM   MEMSV,6,10
        TD   T1,MEMSV,11
        CF   T1
        SM   MEMSV,6,10
        CM   T1,6,10
        BE   *&48
        CM   T1,5,10
        BE   *&24
        B    GAMMA&12
        CM   MEMSO,7,610
        BH   *&24
        B    GAMMA&12
        CM   MEMSO,12,610
        BH   *&24
        B    CAMATR
        CM   MEMSO,25,610
        BNE  GAMMA&12
        B    CAMATR
        DENDBEGIN
```

```
        DORG30000
P       CM   P-18,11,10
        BE   ADDI
        CM   P-18,12,10
        BF   ADDI
        BH   AFFECT
        CM   P-18,8,10
        BE   INVERS
        B    MULTI
AFFECTBD   AF3,P-7
        BNF  *&60,P-6
        SM   P-1,5,10
        MM   P-1,7,610
        S    P-1,99
        SM   P-1,25,10
        AM   P-1,5,10
        TF   C1C2,P-1,11
        AM   P-1,10,10
        M    C1C2,P-1,11
        SF   95
        TF   C1C2,99
        MM   C1C2,12,10
        SF   95
C1C2    DS   *,5
        TF   C1C2,99
        AM   P-1,10,10
        TF   X&6,P-1
        A    X&6,C1C2
        TF   X&11,P-8
X       TFL  ,,611
        SM   X&6,12,10
        SM   C1C2,12,10
        BNE  X
        BB
AF3     TFM  N1,0,10
        BNF  *&60,P-6
        SM   P-1,5,10
        MM   P-1,7,610
        S    P-1,99
        SM   P-1,25,10
        AM   P-1,5,10
        TF   C1C2,P-1,11
        C    N1,0,10
        BE   *&24
        TF   C1C3,C1C2
        AM   P-1,10,10
        M    C1C2,P-1,11
        SF   95
C1C3    DS   *,5,
        TF   C1C2,99
        MM   C1C2,12,10
        SF   95
N1      DS   *,2
        TF   C1C2,99
        AM   P-1,10,10
        TF   Z&6,P-1,11
        A    Z&6,C1C2
        CM   N1,0,10
        BNF  *&72
        AM   N1,1,10
```



```
        TF   P-6,P-1,
        TD   P-7,P-6,
        TF   C1C3,C1C2
        B    AF3&24
        C    C1C2,C1C3
        BE   *&24
        BTM  BUTOIR,16,9
        TF   Z&11,P-1,11
        A    Z&11,C1C2
Z       TFL  ,,611
        SM   Z&6,12,10
        SM   Z&11,12,10
        SM   C1C2,12,10
        BNE  Z
        BB
ADDI    BNF  *&24,P-6
        B    ADDI1
        BNF  *&24,P-7
        B    ADDI1
        SM   P-1,5,10
        SM   P-8,5,10
        C    P-1,P-8,611
        BE   *&24
H       BTM  BUTOIR,17,9
        TF   M,P-1,11
        SM   P-1,10,10
        SM   P-8,10,10
        C    P-8,P-1,611
        BNE  *-60
        SF   95
MN      DS   *,5
        TF   N,P-1,11
        SM   P-1,2,10
        CM   P-1,2,10
        BF   *&24
        BTM  BUTOIR,18,9
        SM   P-1,8,10
        SM   P-1,10,10
        TF   A,P-1,11
        TF   B,P-8,11
        CM   P-18,11,10
        BE   *&36
        TFM  W,0001,8
        B    START
        TFM  W,0011,8
        B    START
TR      DS   *,5,
ADDI1   BNF  *&24,P-6
        B    ADDI2
        SM   P-1,5,10
        MM   P-1,7,10
        S    P-1,99
        SM   P-1,25,10
        B    ADDI&48
ADDI2   BNF  *&24,P-7,
        B    ADDI3
        TF   TR,P-1
        TF   P-1,P-8
        TF   P-8,TR
        B    ADDI&48
```

```
ADDI3 SM   P-1,5,10
      MM   P-1,7,610
      S    P-1,99
      SM   P-8,5,10
      MM   P-8,7,610
      S    P-8,99
      AM   P-1,5,10
      AM   P-8,5,10
      TF   A,P-1,11
      TF   B,P-8,11
      SM   P-13,30,10
      BD   *&60,P-25
      TF   P-13,P-20,611
W1    S    P-13,,6
      TF   TR,P-13,
      B    *&72
      SM   P-13,30,10
      TF   TR,P-13,
      AM   P-13,30,10
      TF   P-13,TR,611
W2    S    P-13,,6
      SM   P-1,8,10
      SM   P-8,8,10
      CM   P-1,2,610
      BE   *&24
      BTM  BUTOIR,18,9
      CM   P-8,2,610
      BNE  *-24
      SM   P-1,7,10
      SM   P-8,7,10
      SM   P-13,7,10
      TFM  P-13,2,610
      SM   P-13,7,10
      C    P-1,P-8,611
      BNH  H
      TF   P-13,P-1,611
      TF   C1C2,P-1,11
      TF   M,P-1,11
      SM   P-1,10,10
      SM   P-8,10,10
      C    P-1,P-8,611
      BNE  H
      TF   P-13,P-1,611
      TF   N,P-1,11
      M    C1C2,P-1,11
      SF   95
      TF   C1C2,99
      TF   W1&11,C1C2
      TF   W2&11,C1C2
      S    TR,C1C2,6
      TF   G,TR,11
      CM   P-18,11,10
      BE   *&36
      TFM  W,0000,8
      B    START
      TFM  W,0010,8
      B    START
```



```
*          SOUS PROGRAMME D@ARITHMETIQUE MATRICIELLE
      DORG*&100
START SF   START-6
      SF   START.8
      SF   START-13
      SF   START-18
      SF   START-20
      M    M,N,
      SF   95
      TF   MN,99,
      BD   LOC1,W-3,
      BD   LOC2,W-2,
      TF   ADD&6,A,
      TF   ADD&11,B,
      TDM  ADD&1,2,,        POUR SOUSTRAIRE
      BD   *&24,W-1,
      TDM  ADD&1,1,,        POUR ADDITIONNER
      BD   ADD,W,
LOC4  TF   MOVE&11,A,
      TF   MOVE&6,G,
      TF   I,MN,
MOVE  TFL
      AM   *-6,12,10
      AM   *-13,12,10
      SM   I,1,10,
      BNE  MOVE
      BD   EXIT,W-2
      TF   ADD&6,G,
ADD   FADD
      AM   *-13,12,10,
      AM   *-6,12,10
      SM   MN,1,10
      BNE  ADD
EXIT  BB
      DORG*-9,
LOC1  BD   SCALAR,W,
      MM   R,12,10
      TF   M14,99,
      TFM  I,0,10
      TF   ACC&23,G
      TFM  J,0,10,
LOOP1 TF   K,N
      TF   MULT&23,A
      MM   J,12,10
      TF   X,99
      M    X,N
      SF   95
      A    MULT&23,99
      TF   MULT&35,B,
      MM   I,12,10
      A    MULT&35,99
MULT  FMUL
      C    K,N
      BE   LOC3
ACC   FADD,99
      AM   MULT&23,12,0
      A    MULT&35,M14
      SM   K,1,10
      BP   MULT
      AM   ACC&23,12,10
```

```
        AM   I,1,10
        C    J,R
        BL   LOOP1
          AM  J,1,10
        C    J,M
        BL   LOOP1-12
        BB
        DORG*-9
        TF   *&18,ACC&6,
        TF   ,99
        B    ACC&36,
        DORG*-3,
        TF   FM&6,A
        TF   KM&11,B
FM      FMUL
        TF   *&18,FM&6,
        TF   ,99
        AM   FM&6,12,10
        SM   MN,1,10
        BNE  FM
        BB
        DORG*-9
LOC2    BD   LOC4,W-1,
        SM   MN,1,10,
        BD   LOC5,W
        MM   MN,1,10
        SF   95
        TF   MN,99,
        MN   N,12,10,
        TF   STORE&11,A,
        TF   STORE&6,G,
        TF   J,N,
        TF   I,M,
STORE   TF
        AM   *-1,12,10
        AM   *-8,99,,          14 N EN 99,
        SM   I,1,10
        BNE  STORE,
        S    STORE&6,MN,,      14 %MN-1□ EN MN,
        SM   J,1,10,
        BNE  STORE-K,
        BB
        DORG*-9,
LOC5    TFM  K,1,8
        TF   TEST&11,A,
        SM   TEST&11,11,10
LOOP2   AM   TEST&11,12,10,
TEST    BNF  FLAG,
        TF   TRANS&11,TEST&11,
        AM   TRANS&11,11,10
        TF   TRANS&18,TRANS&11,
        TF   I,K,
LOOP3   TFM  PI,0,8
DIV     AM   PI,1,10,
        S    I,M,
        BNN  DIV
        M    I,N,
        A    PI,99,
```



```
        A    PI,MN,,           MN-1 EN MN
        C    PI,K
        BE   SEACH
        TF   I,PI
        MM   I,12,10,
        TF   TRANS&23,A,
        A    TRANS&23,99,
        TF   TRANS&30,TRANS&23,
        TF   UNFLAG&6,TRANS&23,
        SM   UNFLAG&6,13,10,
TRANS   TF   TEMP,,,           C%K□ EN TEMP
        TF   ,,,               C%I□ EN C%K□
        TF   ,TEMP,,           TEMP EN C%I□
UNFLAG  CF
        B    LOOP3
        DORG*-3,
        TF   *&18,TEST&11
        SF
SEARCHAM  K,1,10,
        C    K,MN,
        BL   LOOP2,
        BV   *&12,,,
        BB
        DORG*-9,
*           CONSTANTES
G       DS   ,START-21
I       DC   2,0,
X       DC   2,0,
J       DC   2,0,
M14     DS   5
TEMP    DC   2,0,
R       DS   ,START-19,
B       DS   ,START-14,
A       DS   ,START-9,
M       DS   ,START-7,
N       DS   ,START-5,
W       DS   ,START-1,
K       DS   ,START&47
PI      DS   ,START&59,
        DEND
```

REFERENCES

[1] K. SATTLEY : Non-own arrays - ACM Janvier 63.

[2] INGERMANN : Own arrays - ACM Janvier 63.

[3] C. PAIR : Arbres, piles et compilation, Revue française de traitement de l'information 7 (1964) p. 199 - 216

[4] J. ANDRE : Participation à la construction d'un compilateur ALGOL pour 1620 IBM - Thèse de 3ème cycle de Mathématiques 1965 - Centre de Calcul NANCY -

[5] M. CUSEY : Construction d'un compilateur ALGOL pour IBM 1620. - Thèse de 3ème cycle de mathématiques (1964) -

[6] Traitement des procédures : Communication ultérieure du Centre de Calcul de NANCY -

[7] P. NAUR (editor) : Revised report on the algorithmic langage ALGOL 60 ACM Juin 1963.

[8] Brochure IBM 1620 10. 1339. Mai 1962

[9] L. BOLLIET - N. GASTINEL - P. J. LAURENT : Un nouveau langage scientifique : ALGOL - Hermann, Paris 1964.

[10] EPL : European program library (IBM)

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