

FONDS DOCUMENTAIRE
SMP

SOLAR

FFM

Programme de test

MATÉRIEL

MATÉRIEL

MATÉRIEL

MATÉRIEL

MATÉRIEL





TEST OPTION FFM

MANUEL DE FONCTIONNEMENT

SOMMAIRE

	<i>Page</i>
<i>1 - BUT DU PROGRAMME</i>	<i>1.1</i>
<i>2 - CONCEPTION DU TEST</i>	<i>2.1</i>
<i>2.1 - DEFAULT SECTEUR</i>	<i>2.1</i>
<i>2.2 - ORGANISATION DE LA TACHE HARD</i>	<i>2.1</i>
<i>2.3 - RECETTE REC</i>	<i>2.3</i>
<i>2.4 - RECETTE RNS</i>	<i>2.3</i>
<i>2.5 - REMARQUES</i>	<i>2.3</i>
<i>3 - DESCRIPTION DES CLES ACTION ET OUTIL</i>	<i>3.1</i>
<i>3.1 - CLES ACTION</i>	<i>3.1</i>
<i>3.2 - CLES OUTIL</i>	<i>3.45</i>
<i>3.3 - CLES DISPONIBLES AU NIVEAU 2</i>	<i>3.47</i>

1 - BUT DU PROGRAMME

Ce programme a pour but de vérifier le bon fonctionnement des fonctions micro-programmées associées au compilateur FORTRAN (option MI).

Les fonctions testées sont les suivantes :

- boucle DO avec pas égal à 1
- passage des paramètres formels dans l'instruction "subroutine".
- IF arithmétique
- GO TO au-delà de 128 mots
- indexations doubles et triples
- instructions flottant simple sur KSTORE
- schéma de HORNER.

2 - CONCEPTION DU TEST

Ce test est compatible avec les autres tests SOLAR.

La notion de clés est toujours existante.

Chaque clé testant une fonction particulière.

Cependant il existe une différence entre :

- les tests standard qui sont écrits en assembleur

- et ce test qui comprend un module assembleur et des modules FORTRAN.

Pour presque toutes les fonctions, deux sous-programmes FORTRAN sont compilés. Le premier permet l'exécution de la fonction sans l'option MI ; le deuxième permet l'exécution de la fonction avec l'option MI.

Une phase de test est déterminée par une suite de clés données sur le périphérique de dialogue.

La zone réservée au chargement de ces clés est de 40 mots.

On peut donc faire exécuter au maximum :

- 40 clés sans paramètre

- 30 clés avec 10 paramètres

- etc...

toujours de manière à avoir :

nombre de clés + nombre de paramètres \leq 40.

Ce programme fonctionne avec 8 K de mémoire.

2.1 - DEFAULT SECTEUR

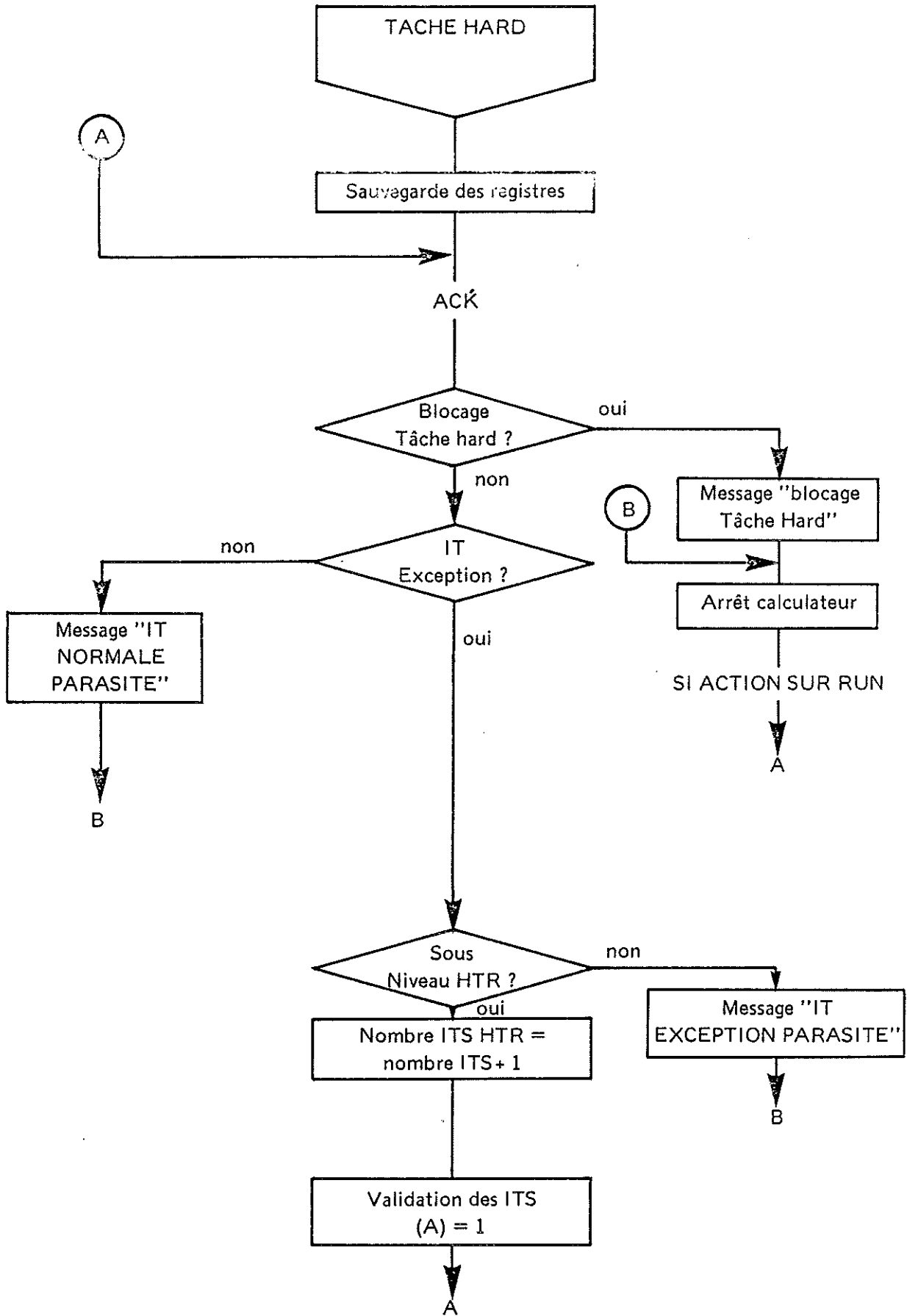
A la relance secteur, le programme édite le message : ** DEFAULT SECTEUR.

Si aucune clé n'était en cours d'exécution, le déroulement reprend à la définition d'une phase de test.

Si une clé était en cours d'exécution, le programme en reprend l'exécution après édition du message : RELANCE CLE EN COURS.

2.2 - ORGANISATION DE LA TACHE HARD

C'est la tâche hard de l'horloge temps réel.



2.3 - RECETTE REC

Elle permet de contrôler le bon fonctionnement de l'option micro-programmée FORMIC. Elle n'existe qu'en mode normal : il n'y a pas de mode debug. Elle assure l'enchaînement suivant :

01	100	test passage de paramètres
02	101	test du DO
03	102	test du GO TO calculé
04	103	test du GO TO au-delà de 128 mots
05	204	test indexation double
06	205	test indexation triple
07	106	test comparaison accu, sommet de pile
08	107	test addition accu, sommet de pile
09	108	test soustraction accu, sommet de pile
0A	109	test division accu, sommet de pile
0B	110	test multiplication accu, sommet de pile
0C	111	test du schéma de HORNER.

2.4 - RECETTE RNS

Elle permet de faire des tests de longue durée. Elle assure l'enchaînement suivant :

01	100	test passage de paramètres
02	101	test du DO
03	102	test du GO TO calculé
04	103	test du GO TO au-delà de 128 mots
05	204	test indexation double
06	205	test indexation triple
07	106	test comparaison accu, sommet de pile
08	107	test addition accu, sommet de pile
09	108	test soustraction accu, sommet de pile
0A	109	test division accu, sommet de pile
0B	110	test multiplication accu, sommet de pile
0C	111	test du schéma de HORNER
0D	BRL 1	* rebouclage à l'infini à la ligne 1.

2.5 - REMARQUES

1) Dans les clés 100, 101, 102, 103, 104, 105, 204 et 205, les sous-programmes FORTRAN (avec et sans option MI) sont exécutées 5000 fois (pour permettre une comparaison entre les temps d'exécution). Ce nombre de tours est visualisé au pupitre du haut.

2) Le nombre de tours de RNS est visualisé sur le pupitre du bas.

3 - DESCRIPTION DES CLES ACTION ET OUTIL

On trouvera dans les pages suivantes la description de toutes les clés action et outil.

Pour chaque clé, une première partie donne tous les renseignements nécessaires à sa mise en oeuvre, ainsi que le but recherché.

A la suite, on trouvera un organigramme simple qui en montre le fonctionnement.

3.1 - CLES ACTION

Une progression est assurée dans les clés action.

La première clé à exécuter est la clé 100 puis vient la clé 101...

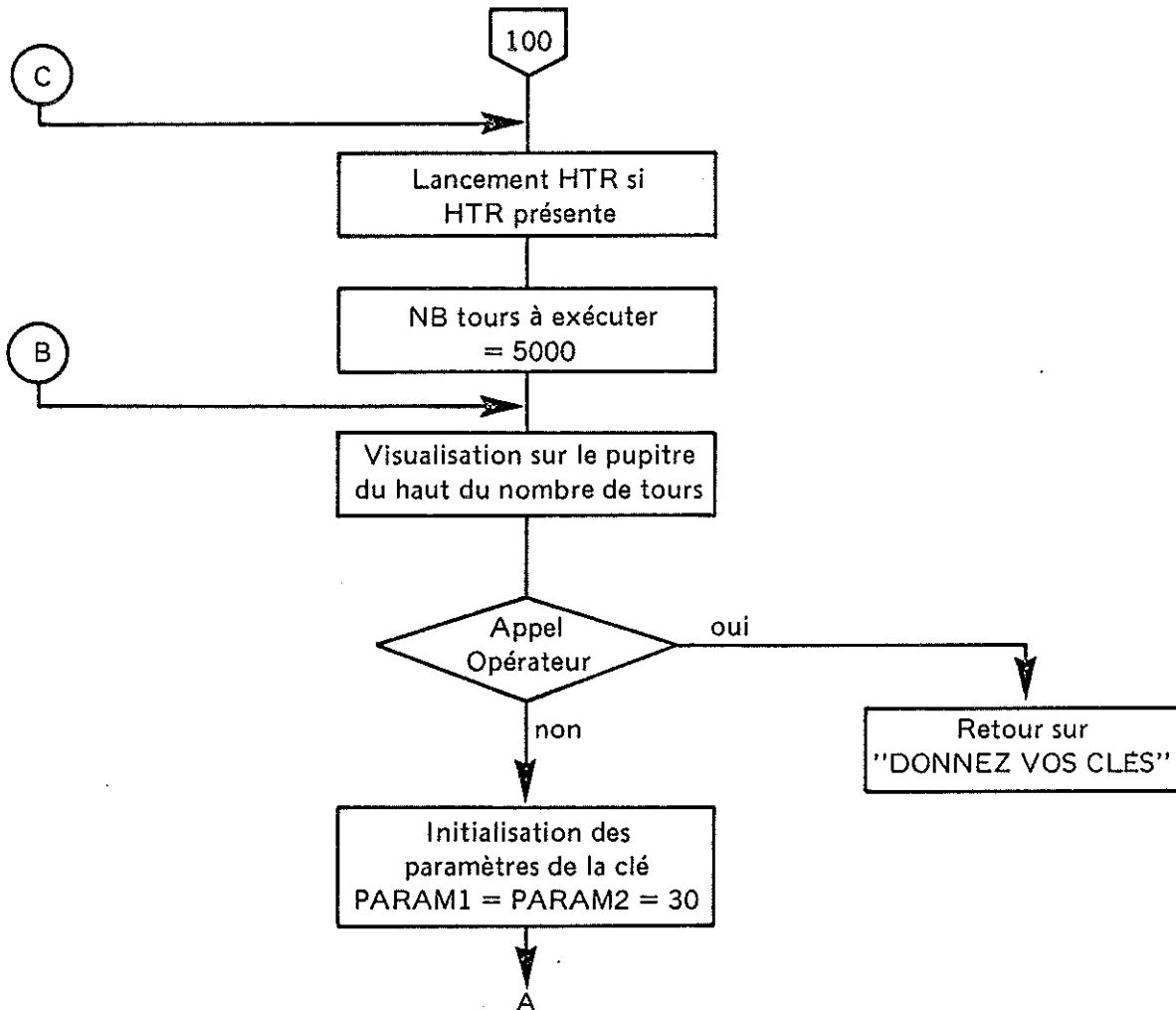
La clé 111.

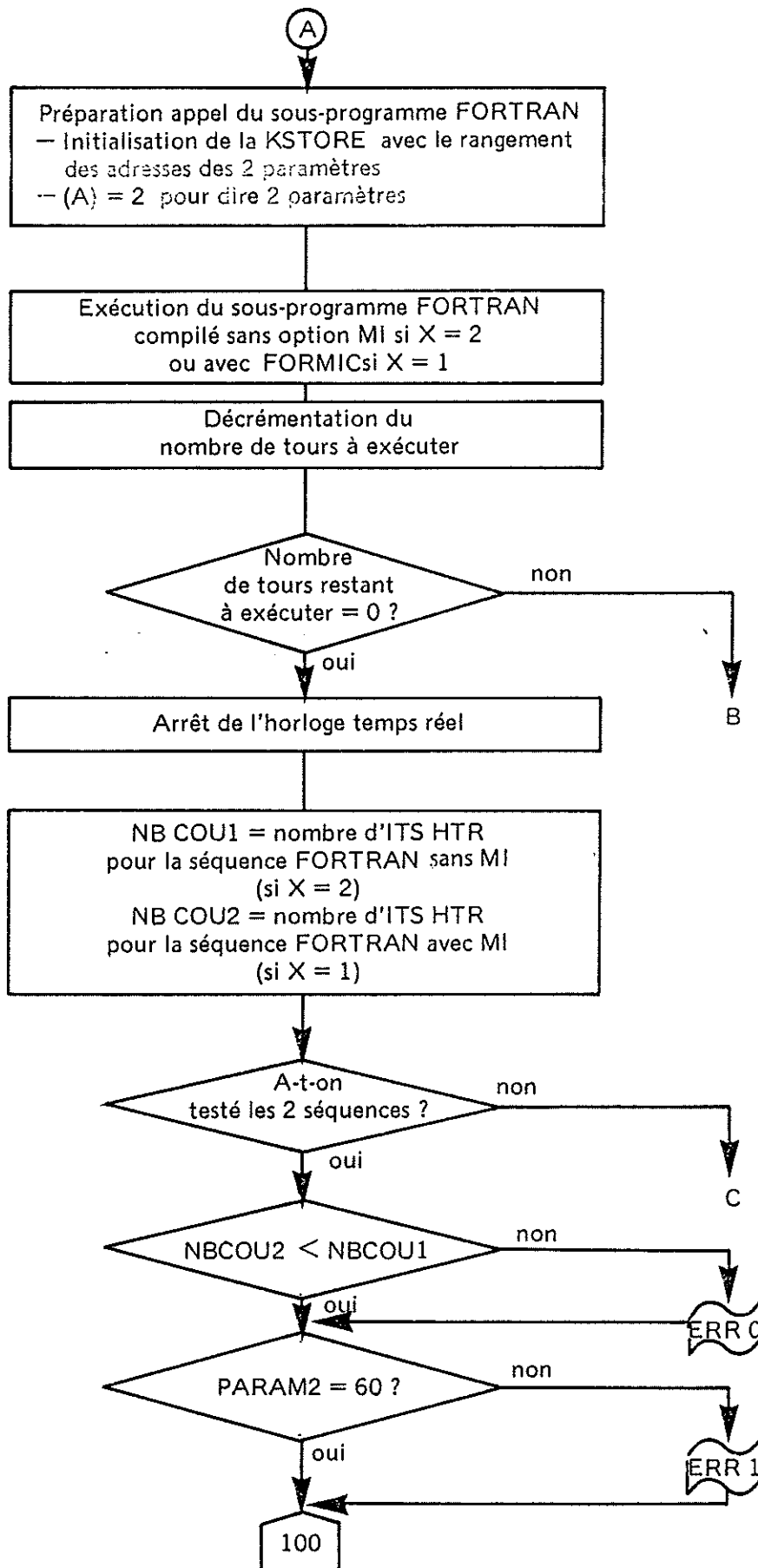
D'une clé sur l'autre, il y a utilisation de l'instruction testée dans la clé précédente.

CLE ACTION	100
Paramètre	non
Clé de PEC	oui
Clé de RNS	oui
<i>BUT</i> : test passage de paramètres.	

2 paramètres sont passés au sous-programme FORTRAN qui charge le deuxième paramètre avec le double du premier.

ORGANIGRAMME DE LA CLE 100.





TEXTE DES SOUS-PROGRAMMES FORTRAN ASSOCIES A LA CLE 100.

```
SUBROUTINE TSTSP (PARM1, PARM2)
  IMPLICIT INTEGER(A-Z)
  PARM2=PARM1#2
  RETURN
END
```

CLE ACTION	101
Paramètre	non
Clé de REC	oui
Clé de RNS	oui
But : test du DO	

CLE 101 : test du DO avec pas égal à 1.

3 paramètres sont passés au sous-programme FORTRAN.

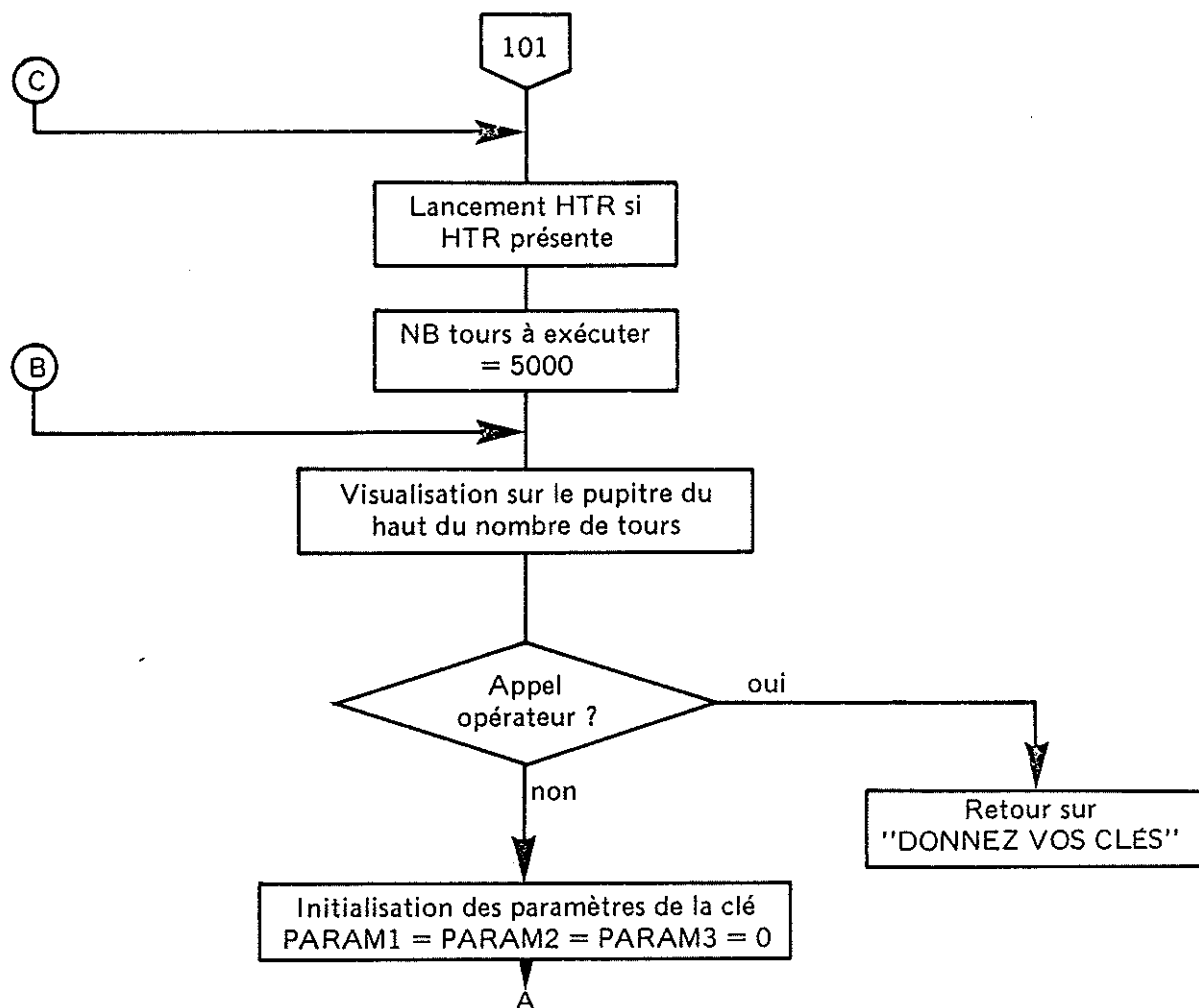
Ces paramètres sont chargés, par le sous-programme FORTRAN, d'une valeur différente, pour distinguer les trois instructions qui permettent le contrôle de fin de boucle DO.

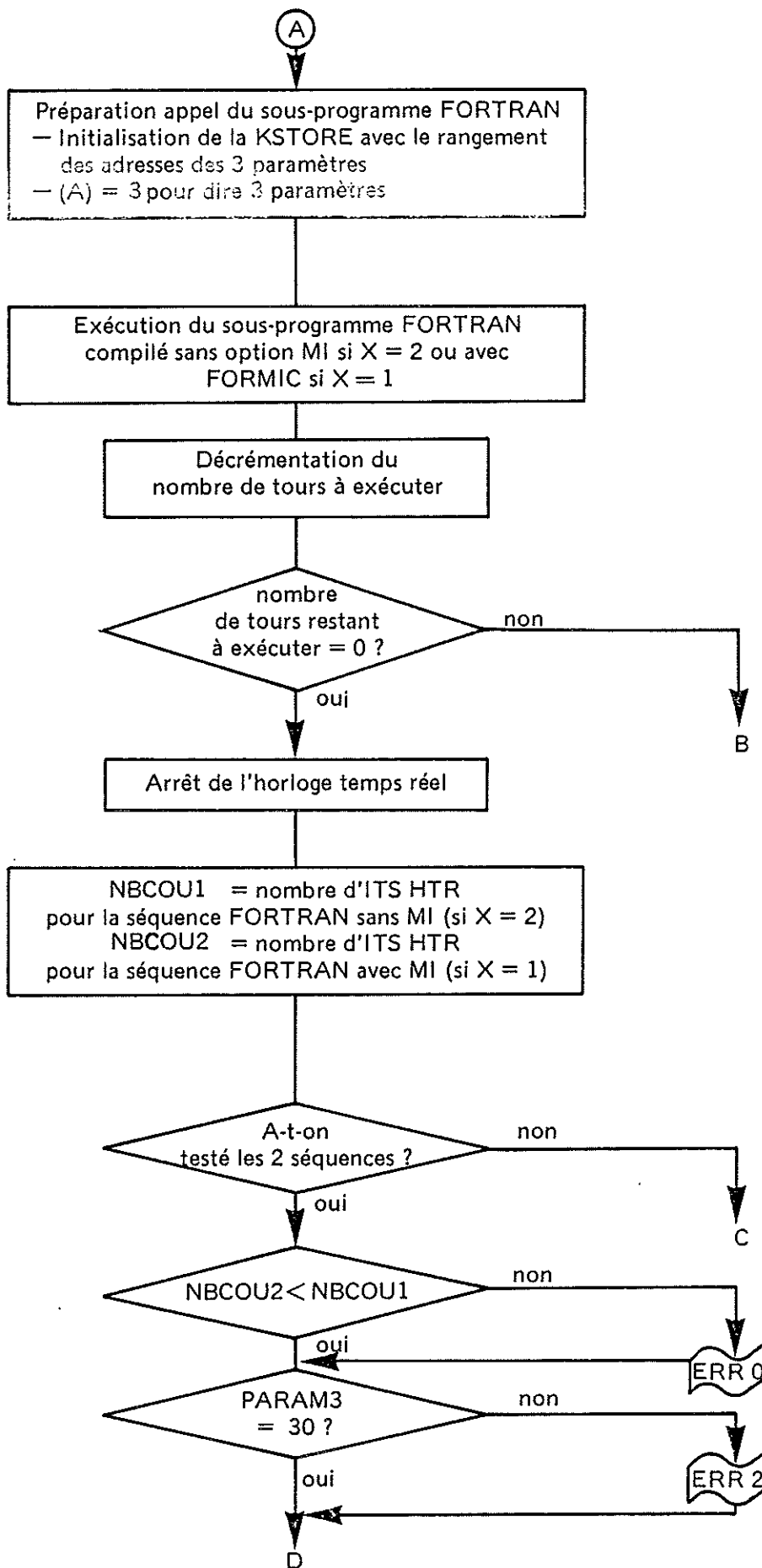
Au retour du sous-programme, le premier paramètre est attendu à 10 (cas du FIND01).

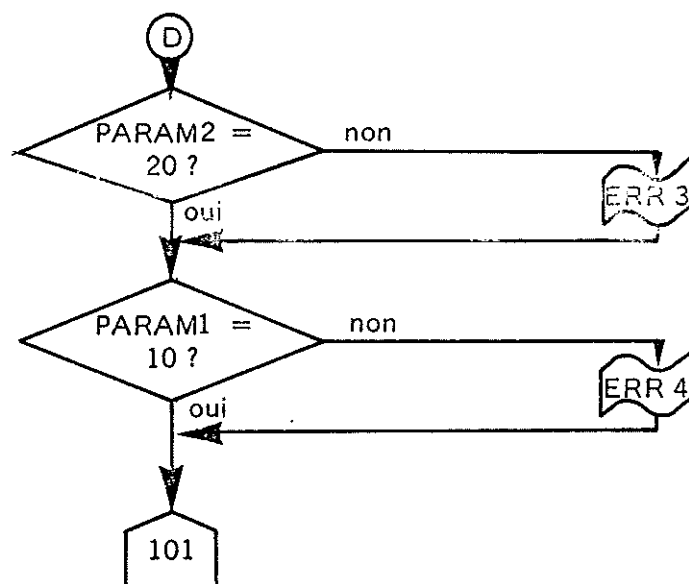
Le deuxième paramètre est attendu à 20 (cas du FIND02)

Le troisième paramètre est attendu à 30 (cas du FIND03).

ORGANIGRAMME DE LA CLE 101.







TEXTE DES SOUS-PROGRAMMES FORTRAN ASSOCIES A LA CLE 101.

```
SUBROUTINE TSTD0 (PARM1,PARM2,PARM3)
IMPLICIT INTEGER (A-Z)
COMMON I
I=10
DO 1 B1=1,I
  PARM1=B1
J=20
DO 2 B2=1,J
  PARM2=B2
DO 3 B3=1,30
  PARM3=B3
RETURN
END
```

CLE ACTION	102
Paramètre	non
Clé de REC	oui
Clé de RNS	oui
But : Test du IF arithmétique.	

CLE 102 : test du GO TO CALCULE

3 paramètres sont passés au sous-programme FORTRAN.

Au retour du sous-programme, le premier paramètre est égal à 10 si le branchement négatif a été correct,

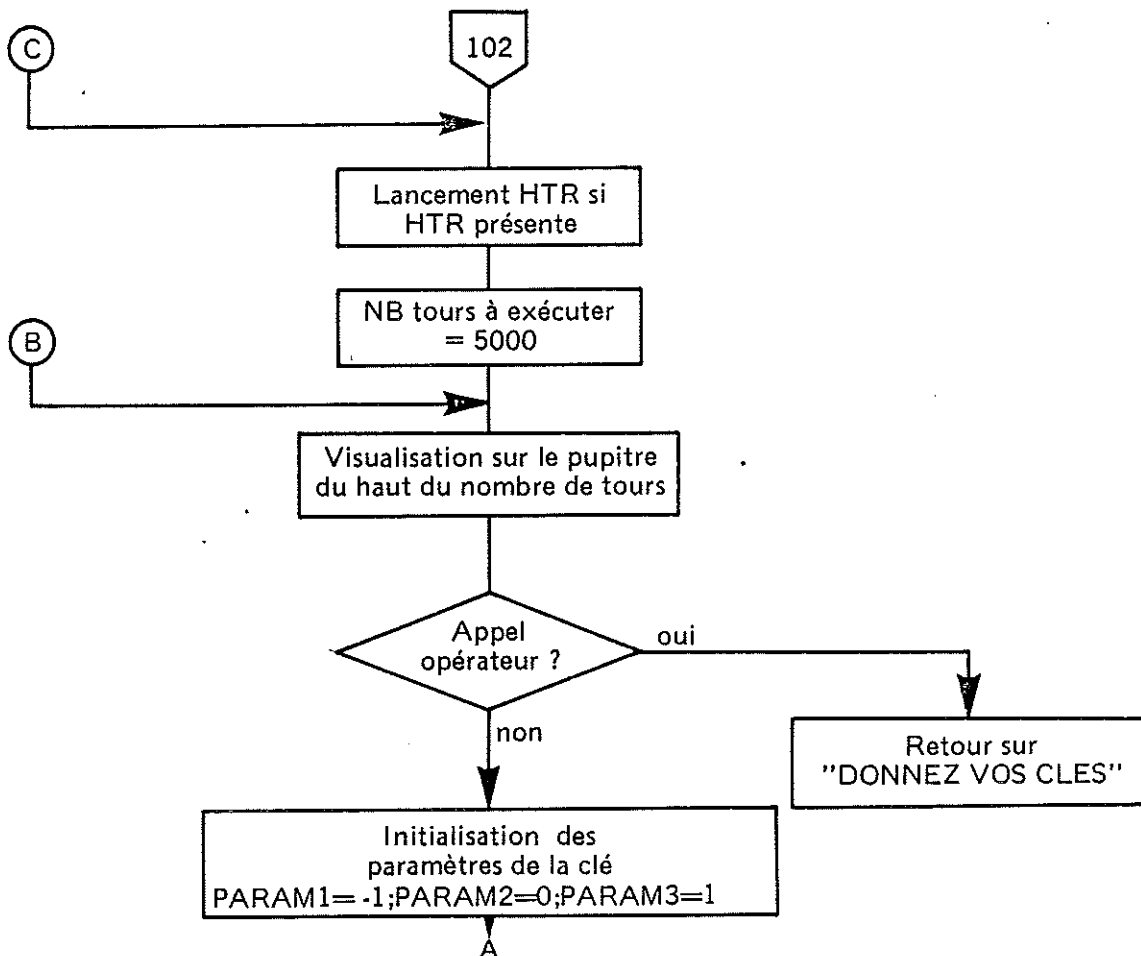
Le deuxième paramètre est égal à 21 si le branchement nul s'est bien passé,

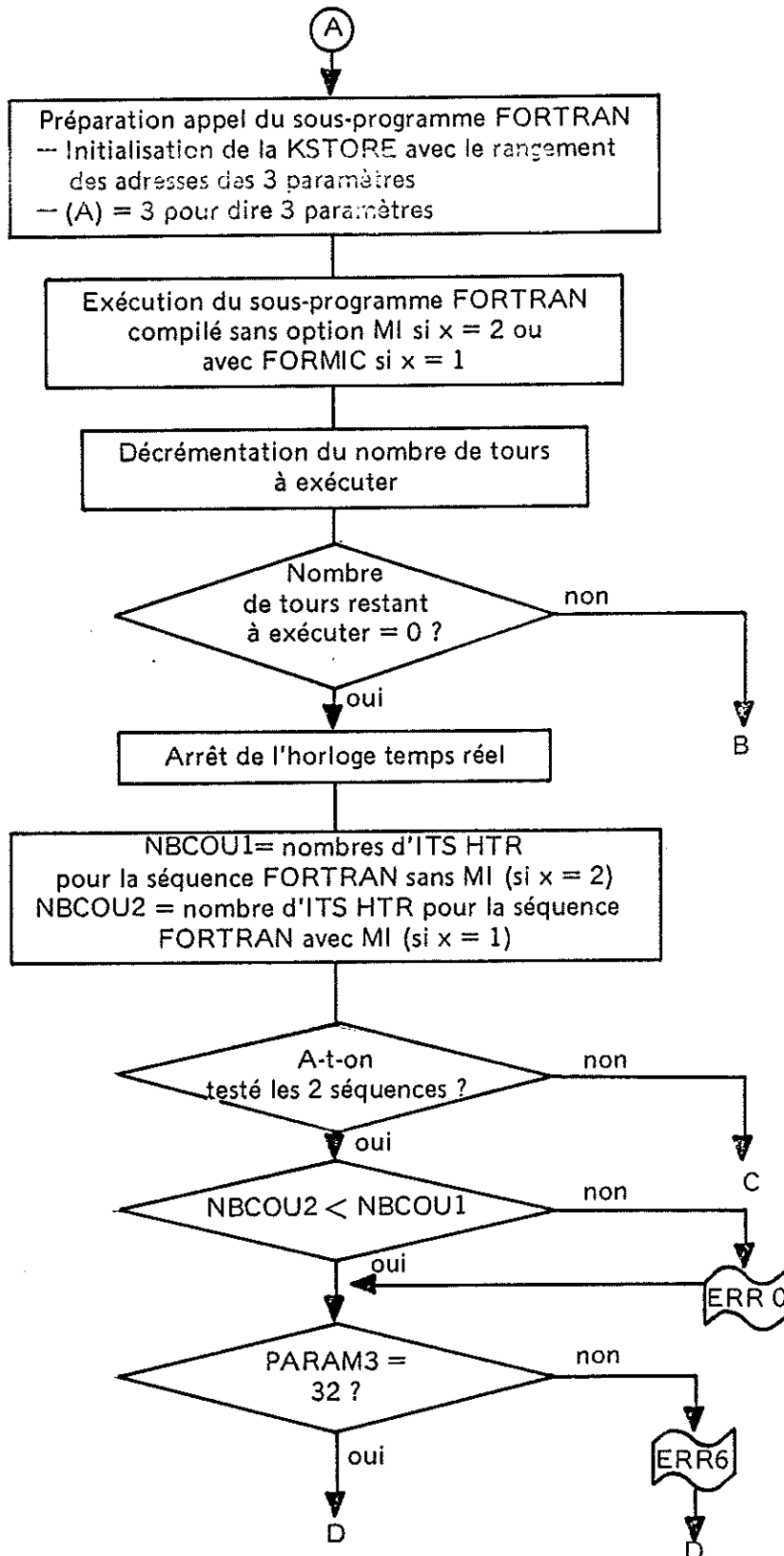
Le troisième paramètre est égal à 32 si le branchement positif a été correct.

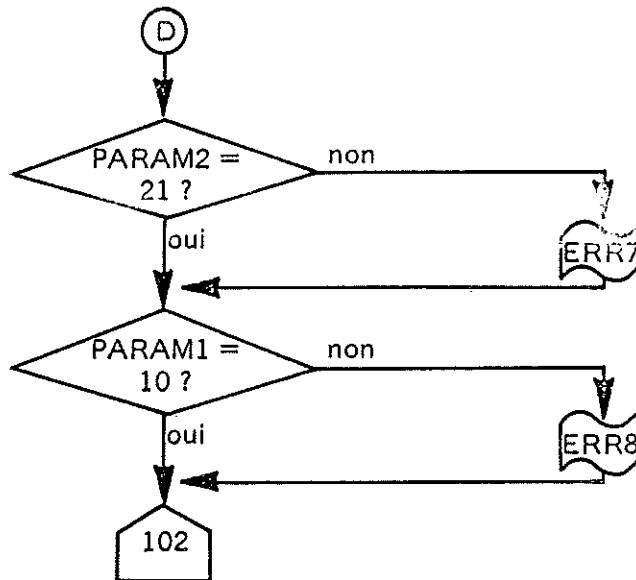
Dans les 3 cas, le paramètre :

- égal à 10, 20 ou 30 correspond au branchement négatif,
- égal à 11, 21 ou 31 correspond au branchement nul,
- égal à 12, 22 ou 32 correspond au branchement positif.

ORGANIGRAMME DE LA CLE 102







TEXTE DES SOUS-PROGRAMMES FORTRAN ASSOCIES A LA CLE 102.

```

SUBROUTINE TSTIF(PARM1,PARM2,PARM3)
IMPLICIT INTEGER(A-Z)
IF (PARM1*2) 10,11,12
10  PARM1=10
   GO TO 13
11  PARM1=11
   GO TO 13
12  PARM1=12
13  IF (PARM2*2) 20,21,22
20  PARM2=20
   GO TO 23
21  PARM2=21
   GO TO 23
22  PARM2=22
23  IF (PARM3*2) 30,31,32
30  PARM3=30
   GO TO 33
31  PARM3=31
   GO TO 33
32  PARM3=32
33  RETURN
   END
  
```

10,20,30 sont les adresses de branchement si l'expression arithmétique est négative.

11,21,31 sont les adresses de branchement si l'expression arithmétique est nulle.

12,22,32 sont les adresses de branchement si l'expression arithmétique est positive.

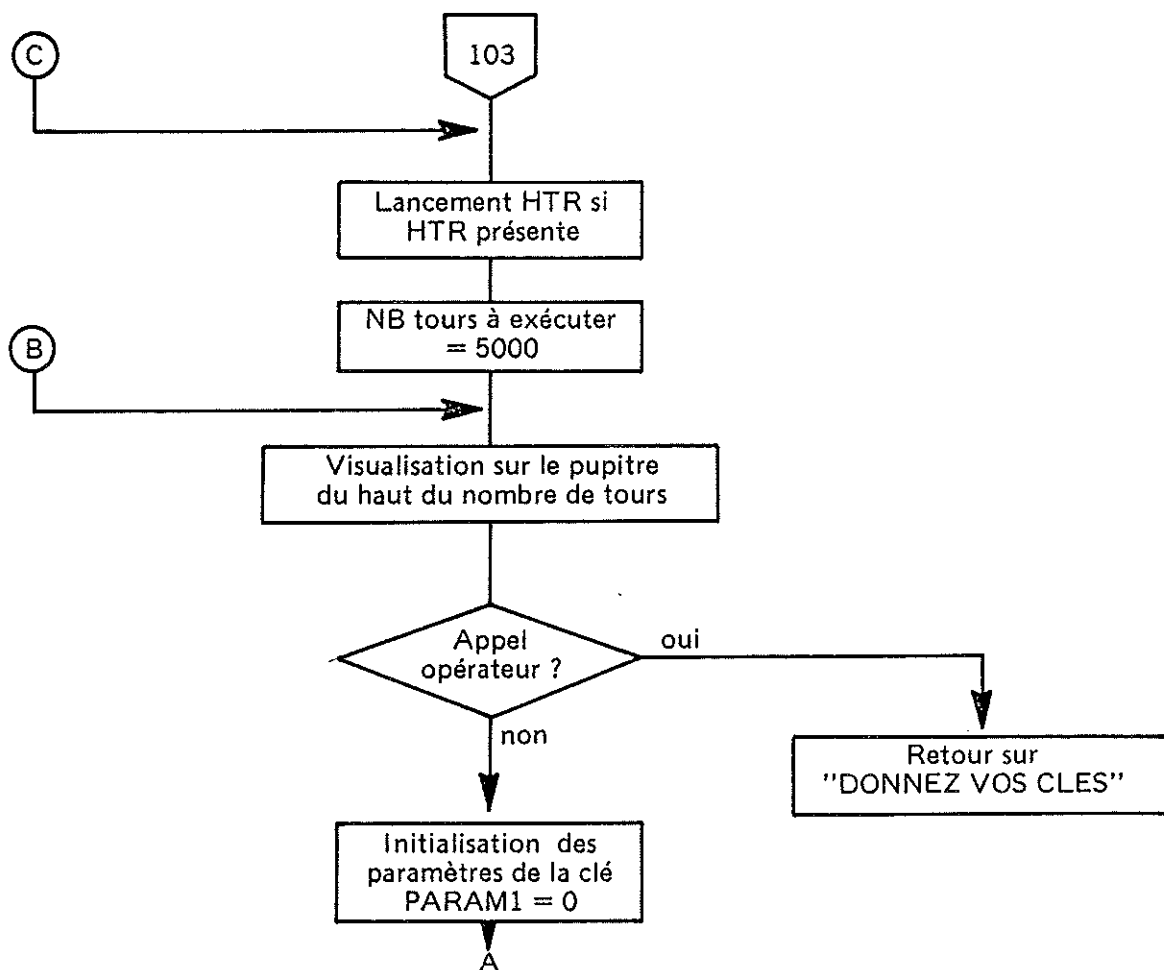
En principe $(\text{PARM1} * 2) = -1 * 2 = -2 < 0$
 $(\text{PARM2} * 2) = 0 * 2 = -0 = 0$
 $(\text{PARM3} * 2) = 1 * 2 = 2 > 0$

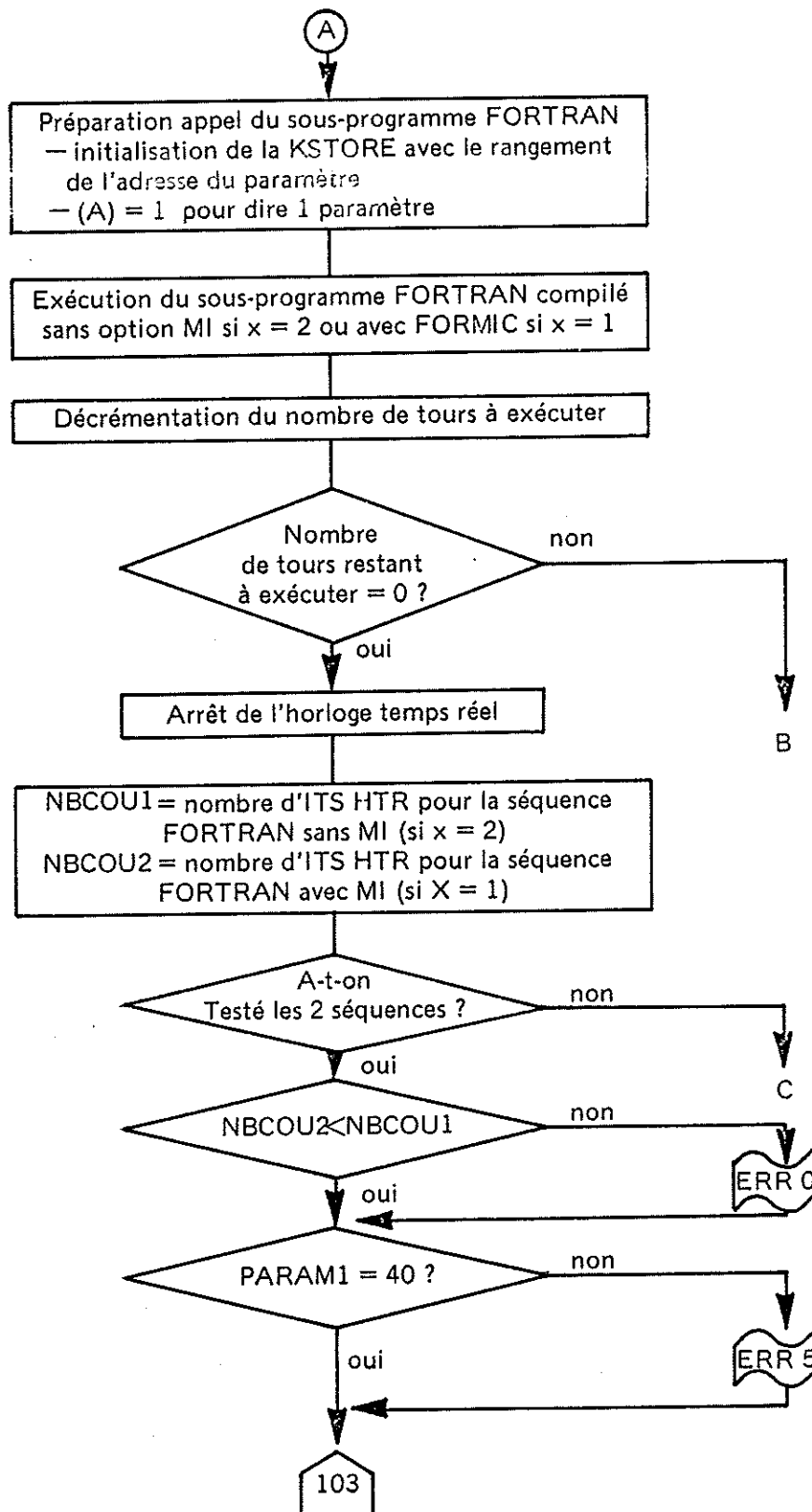
CLE ACTION	103
Paramètre	non
Clé de REC	oui
Clé de PMS	oui
But : test du GO TO au-delà de 128 mots.	

CLE 103 test du GO TO au-delà de 128 Mots.

1 paramètre est passé au sous-programme FORTRAN qui le charge avec 40 si le branchement s'effectue correctement.

ORGANIGRAMME DE LA CLE 103





TEXTE DES SOUS-PROGRAMMES FORTRAN ASSOCIES A LA CLE 103.

```
      SUBROUTINE TSTGO(PARM1)
      IMPLICIT INTEGER(A-Z)
      GO TO 40
      I=I+2
      I=I+2
      I=I+2
      I=I+2
      I=I+2
      I=I+2
      I=I+2
      I=I+2
      I=I+2
      I=I+2
      .
      .
      .
      I=I+2
      PARM1=128
      GO TO 41
40     PARM1=40
41     RETURN
      END
```

} Génération d'un nombre
de mots > 128

Au retour du sous-programme PARAM1 doit être égal à 40. Si PARAM1 = 128, l'instruction GO TO 40 s'est mal exécutée.

CLE ACTION	104
Paramètre	1 { = 0 : tableaux de réels = 1 : tableaux d'entiers = 2 : tableaux de flottant double précision.
Clé de REC	non
Clé de RNS	non
But : Tester l'indexation double.	

Clé 104 test indexation double.

Cette clé a un paramètre qui définit le type des éléments du tableau.

- 104 0 test indexation double sur des tableaux dont les éléments sont des réels.
- 104 1 test indexation double sur des tableaux dont les éléments sont des entiers.
- 104 2 test indexation double sur des tableaux dont les éléments sont des nombres flottant double précision. Dans ce cas, l'option flottant double précision doit avoir été déclarée présente dans le conversationnel. Sinon, il y a édition du message "PAS D'OPTION FLOTTANT DOUBLE PRECISION" sur l'organe de dialogue et retour à "DONNEZ VOS CLES" sans le message "FIN 104 OK".

Dans les cas 104 0 et 104 2, deux tableaux sont chargés par le sous-programme FORTRAN.

TAB (N,2) ÷ la borne étant le paramètre N et N = 2
TAB (10,2) - la borne étant la valeur immédiate 10.

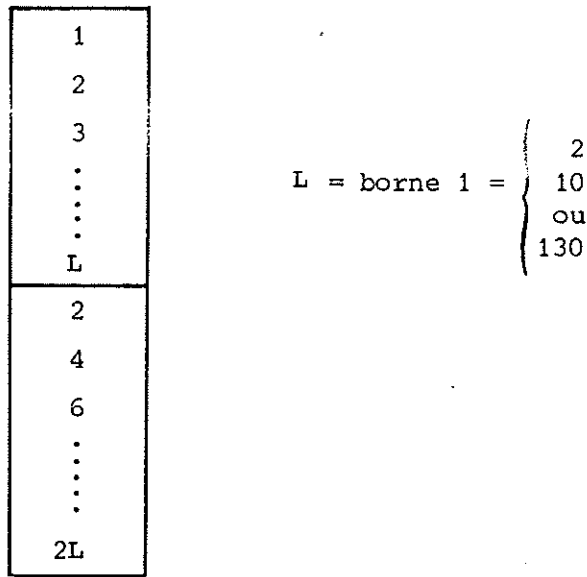
Dans le cas 104 1, trois tableaux sont chargés par le sous-programme FORTRAN.

TAB (N,2)
TAB (10,2)
et TAB (130,2) } idem 104 0 et 104 2

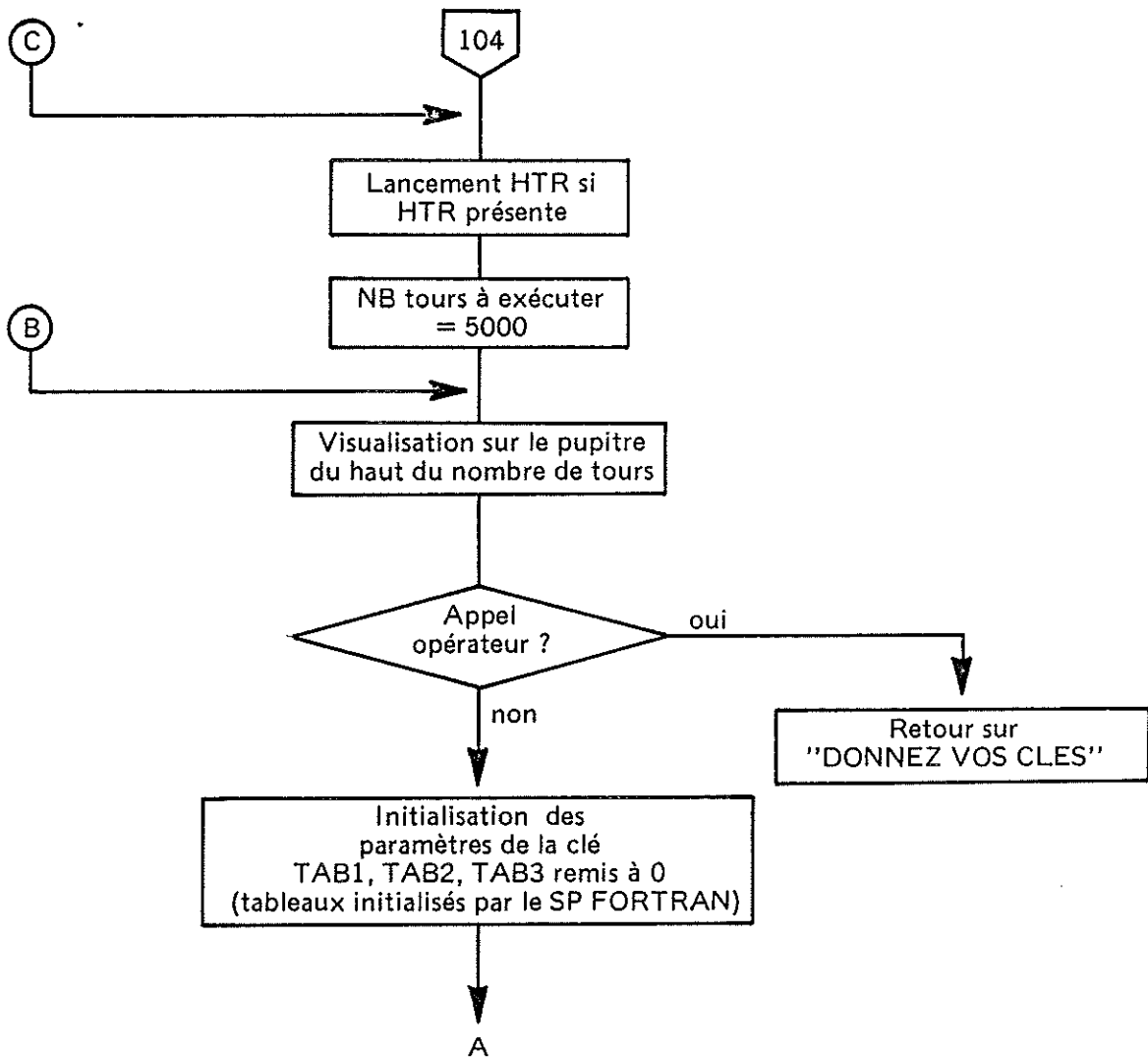
Dans tous les cas, le programme de test vérifie que le sous-programme FORTRAN a effectivement chargé les différents tableaux de la manière suivante :

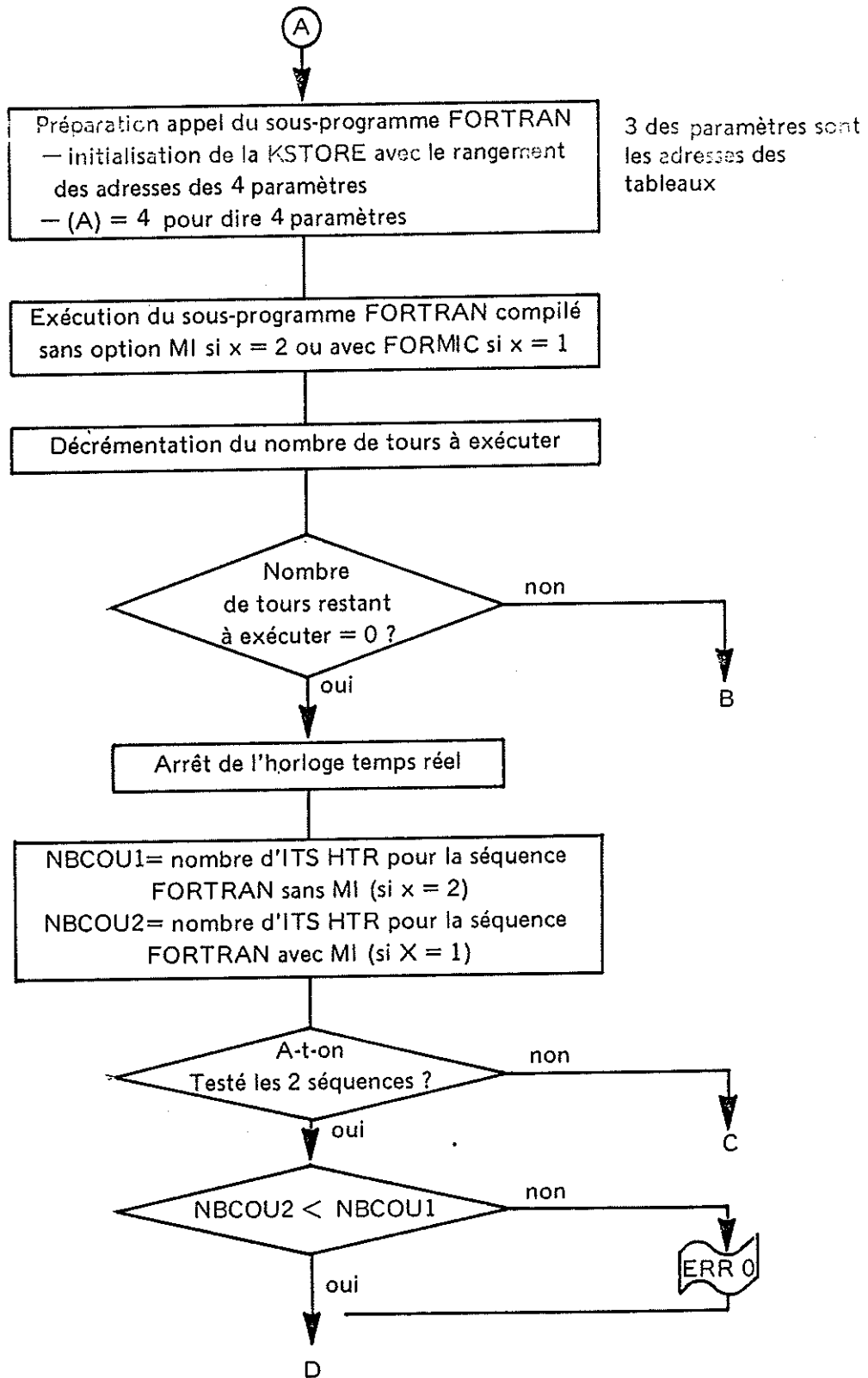
TAB (i, J) = i * J

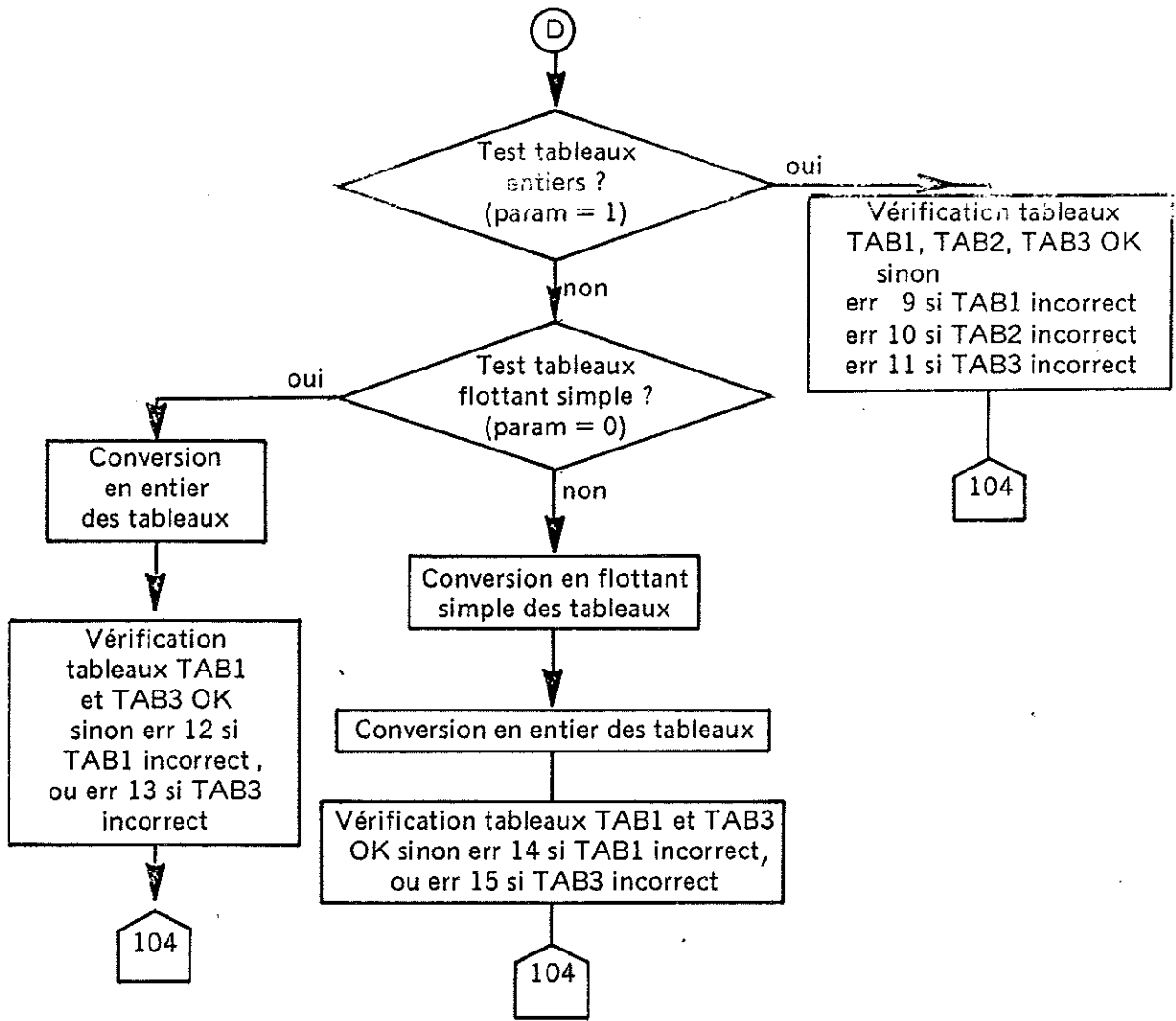
Ce qui se traduit en mémoire par :



Organigramme de la clé 104







) TEXTES DES SOUS-PROGRAMMES FORTRAN ASSOCIES A LA CLE 104.

Pour 104 0 (cas tableaux de réels)

```
      SUBROUTINE TABR(TAB1,TAB2,TAB3,N)
      DIMENSION TAB1(N,2),TAB2(130,2),TAB3(10,2)
      * VARIABLE PARAMETRE
      DO 60 I=1,2
      DO 61 J=1,N
      TAB1 (J,I)=J*I
61      CONTINUE
60      CONTINUE
      * VARIABLE IMMEDIATE
      DO 64 I=1,2
      DO 65 J=1,10
      TAB3(J,I)=J*I
65      CONTINUE
64      CONTINUE
      RETURN
      END
```

Pour 104 1 (cas tableaux d'entiers)

```
      SUBROUTINE TABI(TAB1,TAB2,TAB3,N)
      IMPLICIT INTEGER(A-Z)
      DIMENSION TAB1(N,2),TAB2(130,2),TAB3(10,2)
      * VARIABLE PARAMETRE
      DO 50 I=1,2
      DO 51 J=1,N
      TAB1 (J,I)=J*I
51      CONTINUE
50      CONTINUE
      * VARIABLE SUP 128
      DO 52 I=1,2
      DO 53 J=1,130
      TAB2(J,I)=I*J
53      CONTINUE
52      CONTINUE
      * VARIABLE IMMEDIATE
      DO 54 I=1,2
      DO 55 J=1,10
      TAB3(J,I)=J*I
55      CONTINUE
54      CONTINUE
      RETURN
      END
```

Pour 104 2 (Cas tableaux flottant double précision)

```
      SUBROUTINE TABF(TAB1,TAB2,TAB3,N)
      IMPLICIT INTEGER(A-S)
      IMPLICIT DOUBLE PRECISION(T-Z)
      DIMENSION TAB1(N,2)TAB2(130,2),TAB3(10,2)
      *VARIABLE PARAMETRE
        DO 70 I=1,2
        DO 71 J=1,N
        TAB1(J,I)=J*I
      71 CONTINUE
      70 CONTINUE
      *VARIABLE IMMEDIATE
        DO 74 I=1,2
        DO 75 J=1,10
        TAB3(J,I)=J*I
      75 CONTINUE
      74 CONTINUE
      RETURN
      END
```

Dans les cas 104 0 et 104 2, seuls les tableaux TAB1 et TAB3 sont chargés par les sous-programmes FORTRAN.

Dans le cas 104 1, TAB1, TAB2 et TAB3 sont chargés par les sous-programmes FORTRAN.

CLÉ ACTION	105
Paramètre	1 { = 0 : tableaux de réels = 1 : tableaux d'entiers = 2 : tableaux de flottant double précision
Clé de REC	non
Clé de RNS	non
<i>But</i> : Tester l'indexation triple.	

CLÉ 105 test indexation triple

Cette clé a un paramètre qui définit le type des éléments du tableau.

- 105 0 test indexation triple sur des tableaux dont les éléments sont des réels.
- 105 1 test indexation triple sur des tableaux dont les éléments sont des entiers.
- 105 2 test indexation triple sur des tableaux dont les éléments sont des nombres flottant double précision. Dans ce cas, l'option flottant double précision doit avoir été déclarée présente dans le conversationnel. Sinon, il y a édition du message "PAS D'OPTION FLOTTANT DOUBLE PRECISION" sur l'organe de dialogue et retour à "DONNEZ VOS CLES" sans le message "FIN 105 OK".

Dans les cas 105 0 et 105 2 deux tableaux sont chargés par le sous-programme FORTRAN. 1) TAB(2,N,2). La borne étant le paramètre N et N = 2. 2) TAB(2,10,2). La borne étant la valeur immédiate 10.

Dans le cas 105 1, trois tableaux sont chargés par le sous-programme FORTRAN.

TAB (2,N,2)
TAB (2,10,2)
et TAB (2,130,2) } idem 105 0 et 105 2

Dans tous les cas, le programme de test vérifie que le sous-programme FORTRAN a effectivement chargé les différents tableaux de la manière suivante.

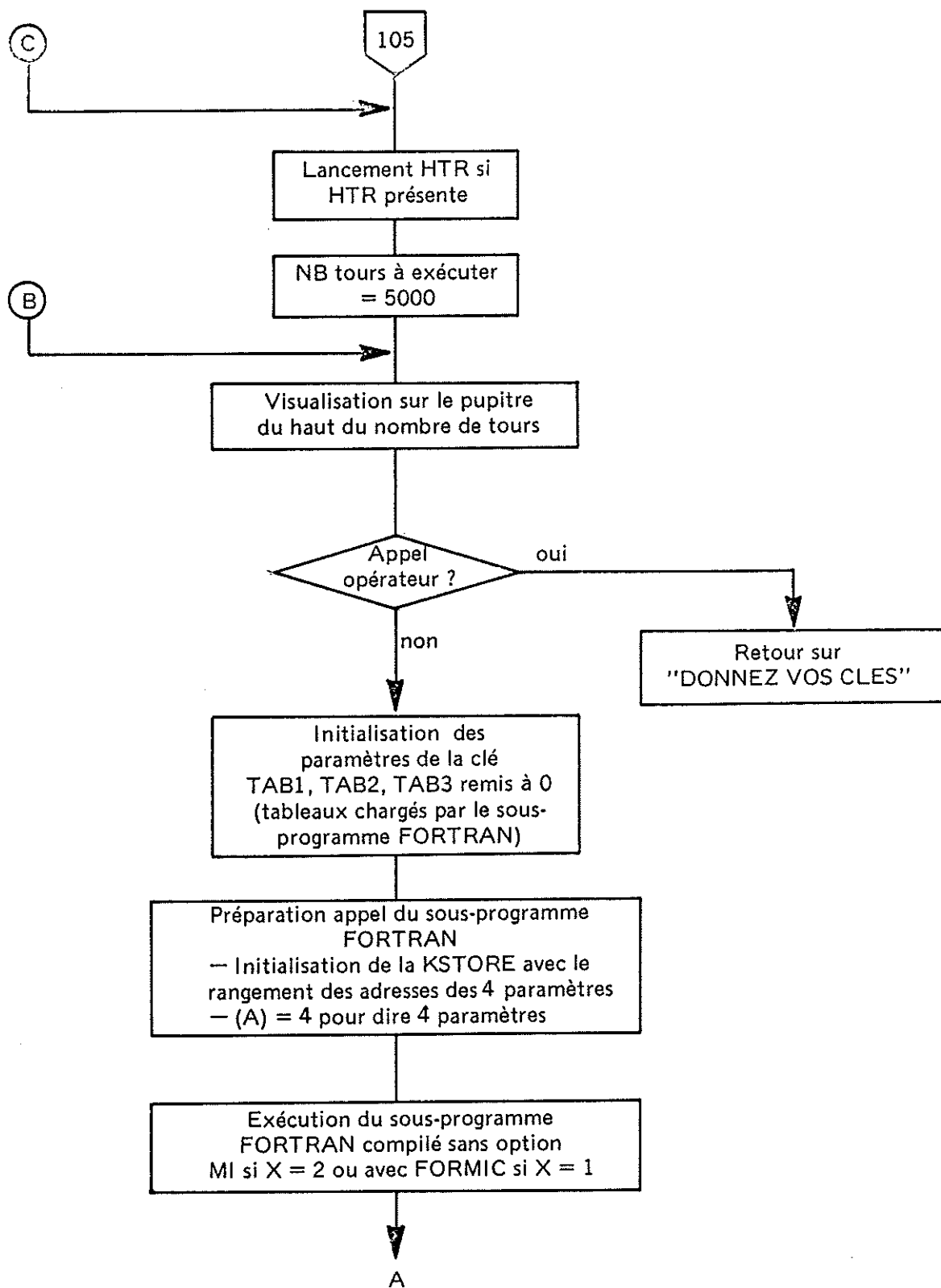
TAB (i,j,k) = J*I*K

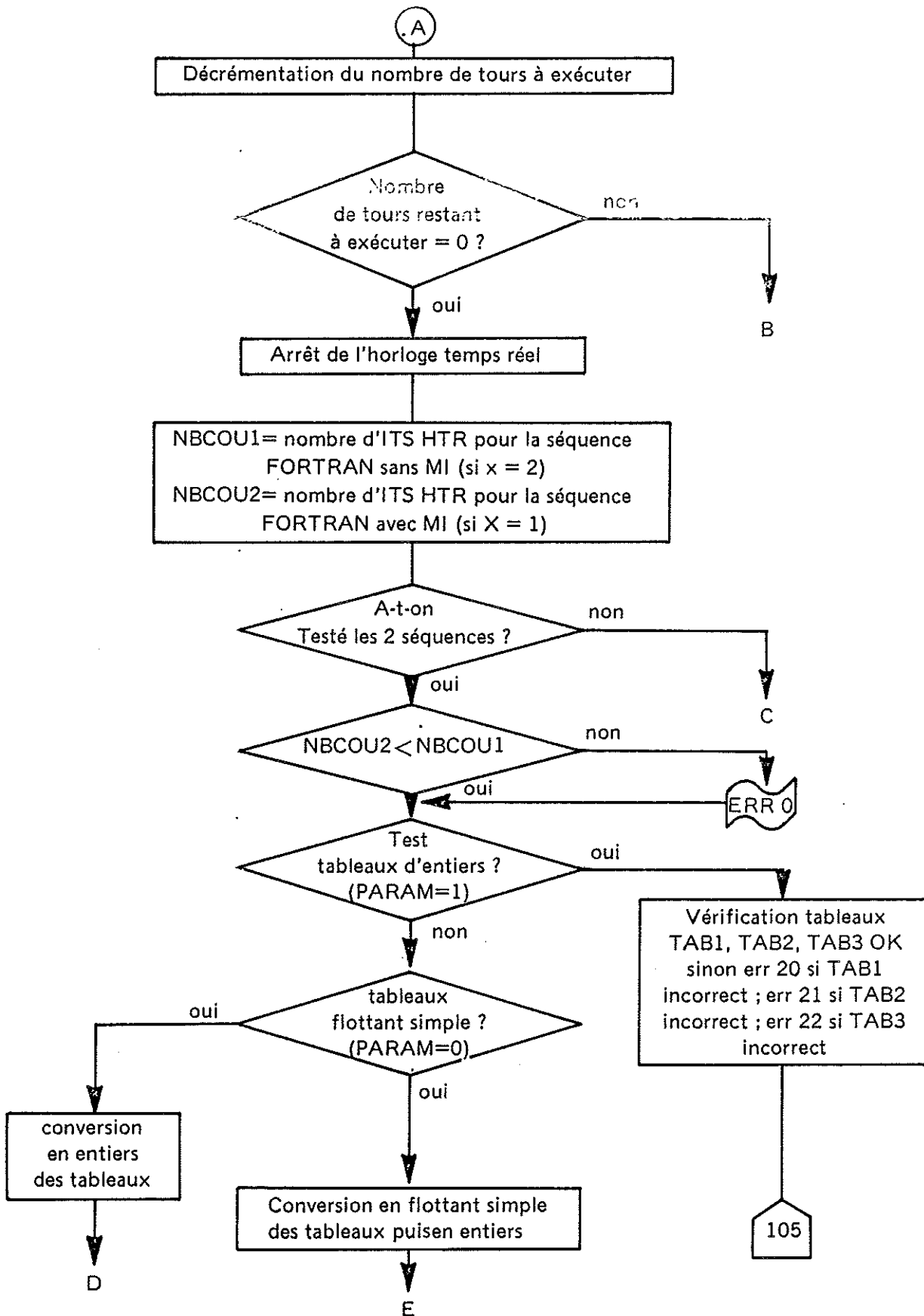
Ce qui se traduit en mémoire par :

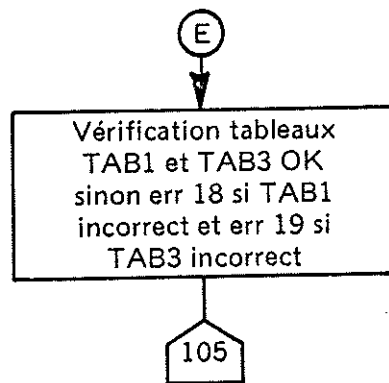
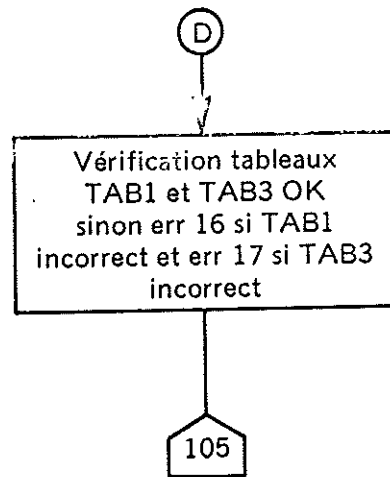
1
2
2
4
3
6
.
.
.
L
2 L
2
4
4
8
6
12
.
.
.
2L
4L

$$L = \text{borne} = \left\{ \begin{array}{l} 2 \\ 10 \\ \text{ou} \\ 130 \end{array} \right.$$

Organigramme de la clé 105







TEXTES DES SOUS-PROGRAMMES FORTRAN ASSOCIES A LA CLE 105.

Si 105 0 (cas tableaux réels)

```

SUBROUTINE TABTR(TAB1,TAB2,TAB3,N)
  DIMENSION TAB1(2,N,2),TAB2(2,130,2),TAB3(2,10,2)
  *VARIABLE PARAMETRE
    DO 90 K=1,2
    DO 91 J=1,N
    DO 92 I=1,2
    TAB1(I,J,K)=J*I*K
  92 CONTINUE
  91 CONTINUE
  90 CONTINUE
  *VARIABLE IMMEDIATE
    DO 93 K=1,2
    DO 94 J=1,10
    DO 95 I=1,2
    TAB3(I,J,K)=J*I*K
  95 CONTINUE
  94 CONTINUE
  93 CONTINUE
  RETURN
  END

```

Si 105 1 (Cas tableaux entiers)

```

SUBROUTINE TABTI(TAB1,TAB2,TAB3,N)
  IMPLICIT INTEGER(A-Z)
  DIMENSION TAB1(2,N,2),TAB2(2,130,2),TAB3(2,10,2)
  *VARIABLE PARAMETRE
    DO 80 K=1,2
    DO 81 J=1,N
    DO 82 I=1,2
    TAB1(I,J,K)=J*I*K
  82 CONTINUE
  81 CONTINUE
  80 CONTINUE
  *VARIABLE SUP 128
    DO 83 K=1,2
    DO 84 J=1,130
    DO 85 I=1,2
    TAB2(I,J,K)=I*J*K
  85 CONTINUE
  84 CONTINUE
  83 CONTINUE
  *VARIABLE IMMEDIATE
    DO 86 K=1,2
    DO 87 J=1,10
    DO 88 I=1,2
    TAB3(I,J,K)=J*I*K

```

```
88 CONTINUE
87 CONTINUE
86 CONTINUE
  RETURN
  END
```

Si 105 2 (*Cas tableaux flottant double précision*)

```
  SUBROUTINE TABTF(TAB1,TAB2,TAB3,N)
  IMPLICIT INTEGER(A-S)
  IMPLICIT DOUBLE PRECISION(T-Z)
  DIMENSION TAB1(2,N,2),TAB2(2,130,2),TAB3(2,10,2)
  *VARIABLE PARAMETRE
    DO 100 K=1,2
    DO 101 J=1,N
    DO 102 I=1,2
      TAB1(I,J,K)=J*I*K
102 CONTINUE
101 CONTINUE
100 CONTINUE
  *VARIABLE IMMEDIATE
    DO 103 K=1,2
    DO 104 J=1,10
    DO 105 I=1,2
      TAB3(I,J,K)=J*I*K
105 CONTINUE
104 CONTINUE
103 CONTINUE
  RETURN
  END
```

CLE ACTION	106
Paramètre	non
Clé de REC	oui
Clé de RNS	oui
But : test instruction flottant simple sur KSTORE. test comparaison accu, sommet de pile.	

CLE 106 Test instruction flottant simple sur KSTORE

- Test comparaison ACCU, sommet de pile.

Le test vérifie que le positionnement des indicateurs V et C est correct après l'instruction de comparaison.

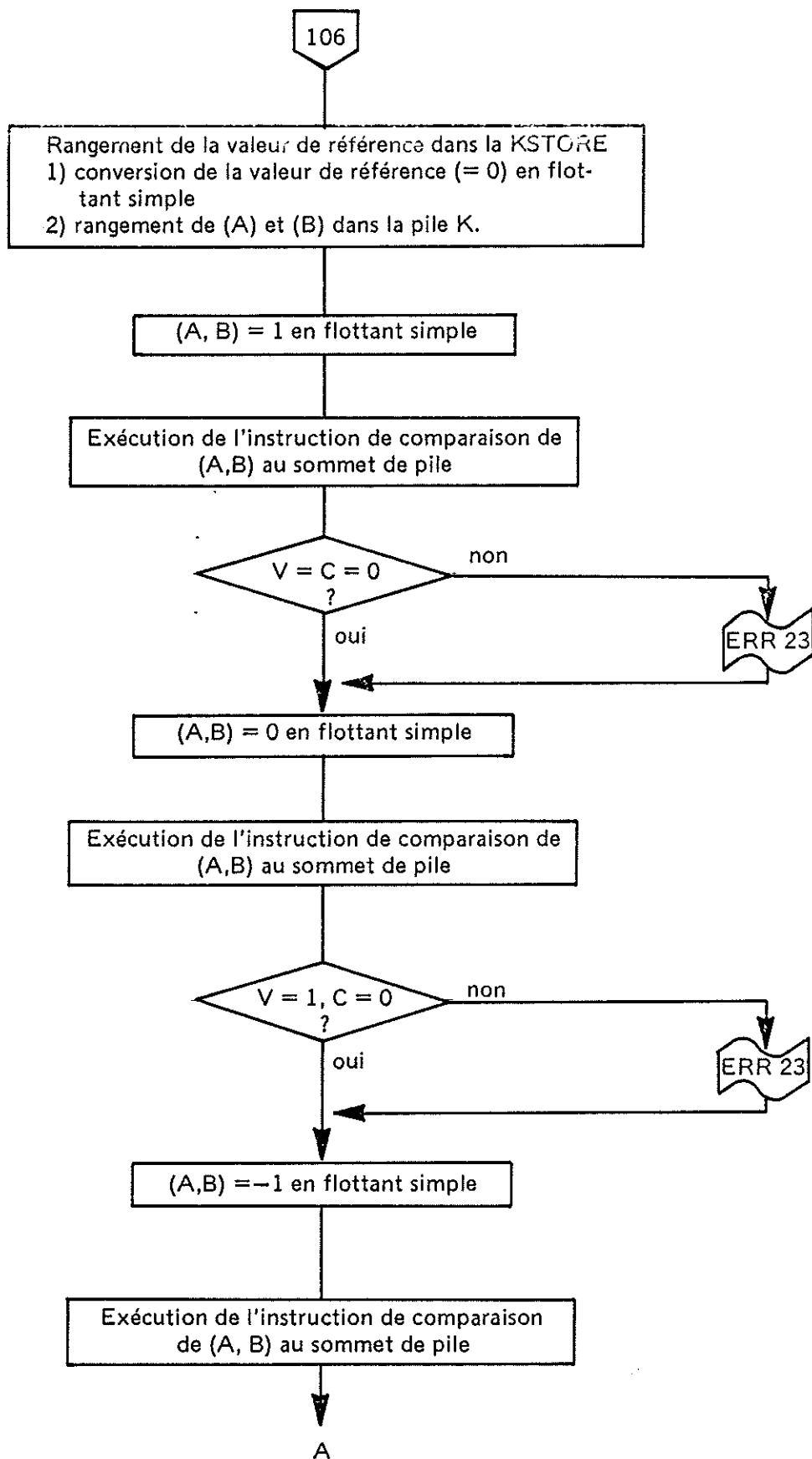
V = 0, C = 1 si AB < sommet de pile

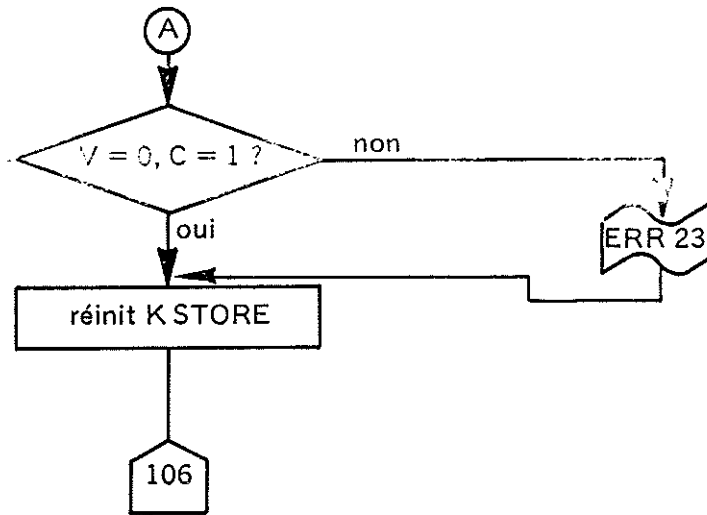
V = 1, C = 0 si AB = sommet de pile

V = 0, C = 1 si AB > sommet de pile

Dans cette clé, il n'y a pas de module FORTRAN.

Organigramme de la clé 106





CLE ACTION	107
Paramètre	non
Clé de REC	oui
Clé de PMS	oui
But : test addition accu, sommet de pile	

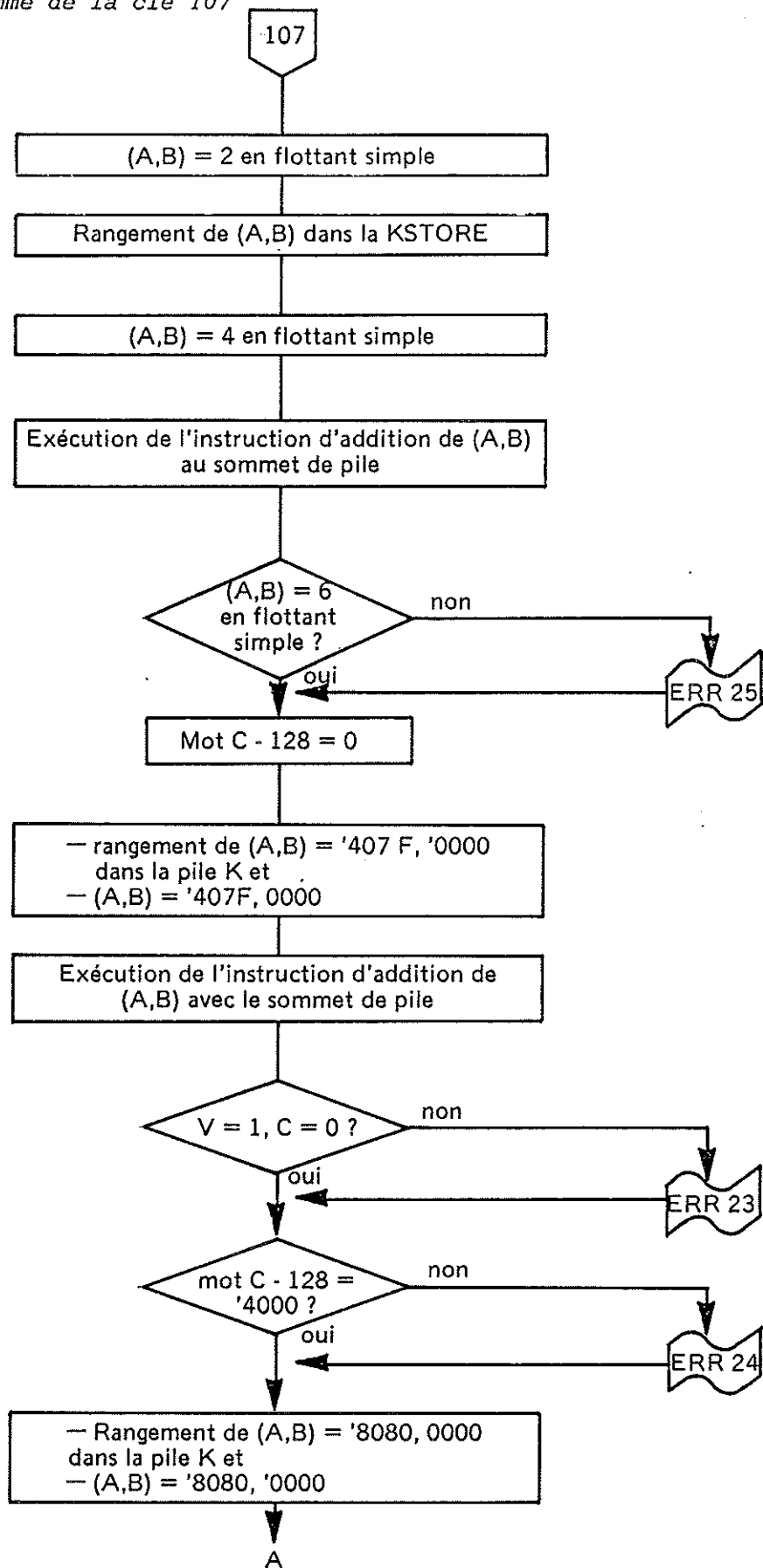
CLE 107 test instruction flottant simple sur KSTORE

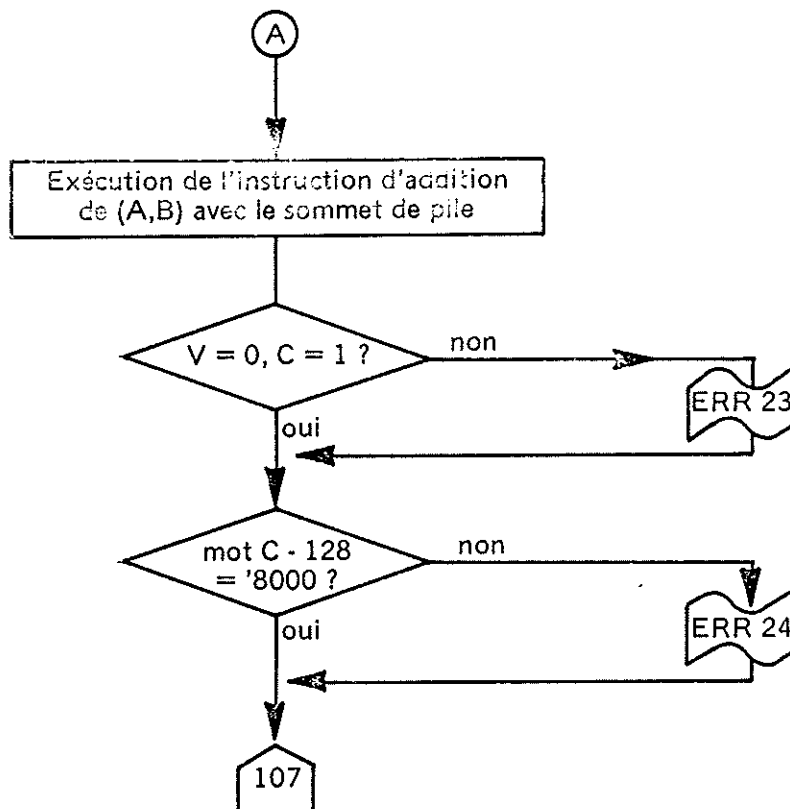
- Test addition accu, sommet de pile.

Le test vérifie que le résultat de l'opération est correcte.
Pour cela, il effectue $(AB) + (\text{sommet de pile})$ avec $(AB) = 4$
et $(\text{sommet de pile}) = 2$ et vérifie que $(AB) = 6$.

D'autre part, il vérifie que le positionnement des indicateurs
V et C dans ST et dans le mot C-128 est correct.

Organigramme de la clé 107





CLE ACTION	108
Paramètre	non
Clé de REC	oui
Clé de RNS	oui
But : test soustraction accu, sommet de pile.	

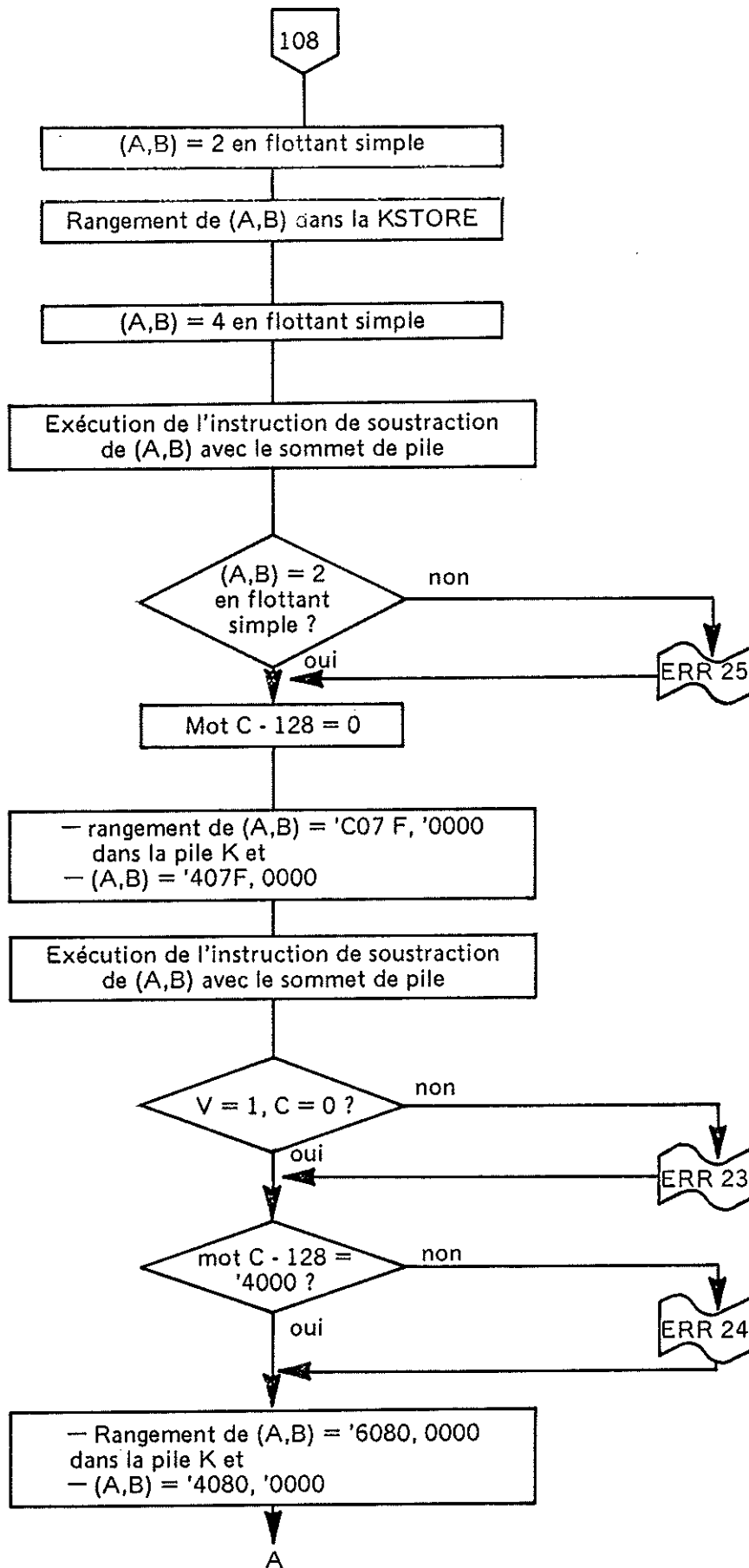
CLE 108 test instruction flottant simple sur KSTORE

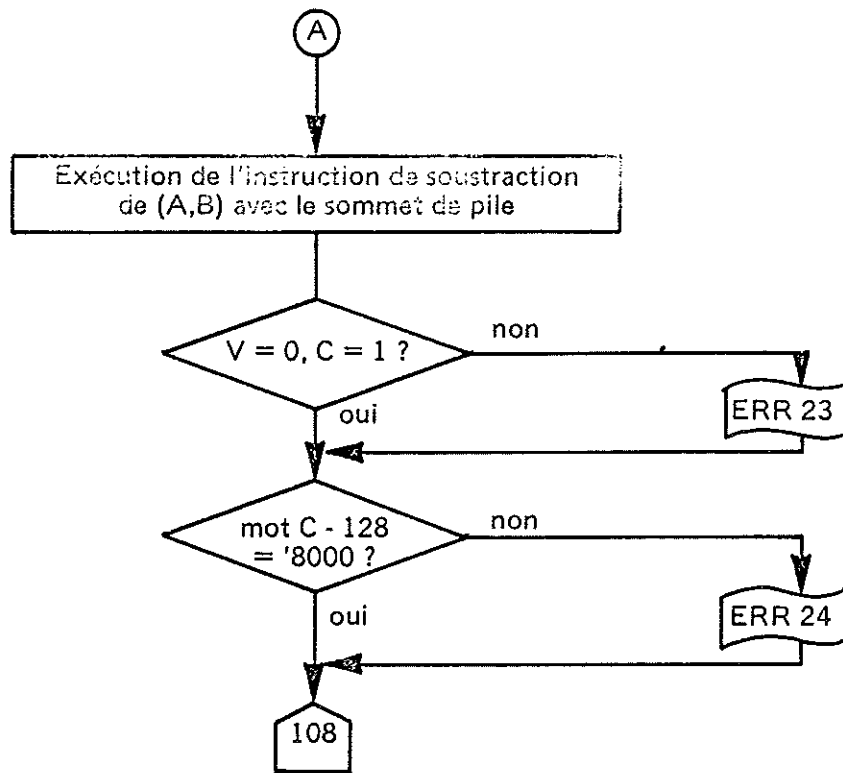
- Test soustraction Accu, sommet de pile.

Le test vérifie que le résultat de l'opération est correcte.
Pour cela il effectue $(AB) - (\text{sommet de pile})$ avec $(AB) = 4$ et
 $(\text{sommet de pile}) = 2$ et vérifie que $(AB) = 2$.

D'autre part, il vérifie que le positionnement des indicateurs
V et C dans ST et dans le mot C-128 est correct.

Organigramme de la clé 108





CLE ACTION	109
Paramètre	non
Clé de REC	oui.
Clé de RMS	oui
But : test division accu, sommet de pile.	

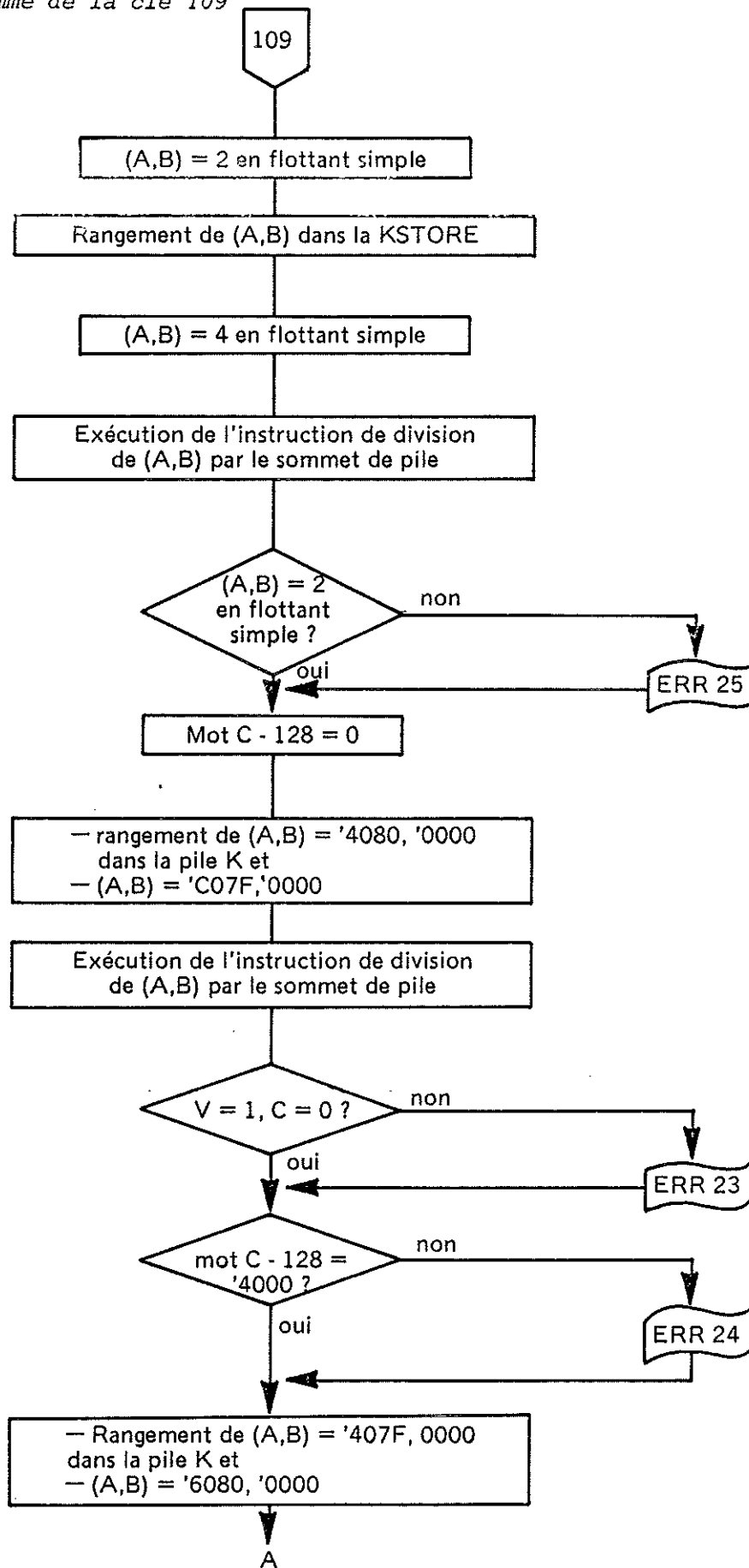
CLE 109 test instruction flottant simple sur KSTORE.

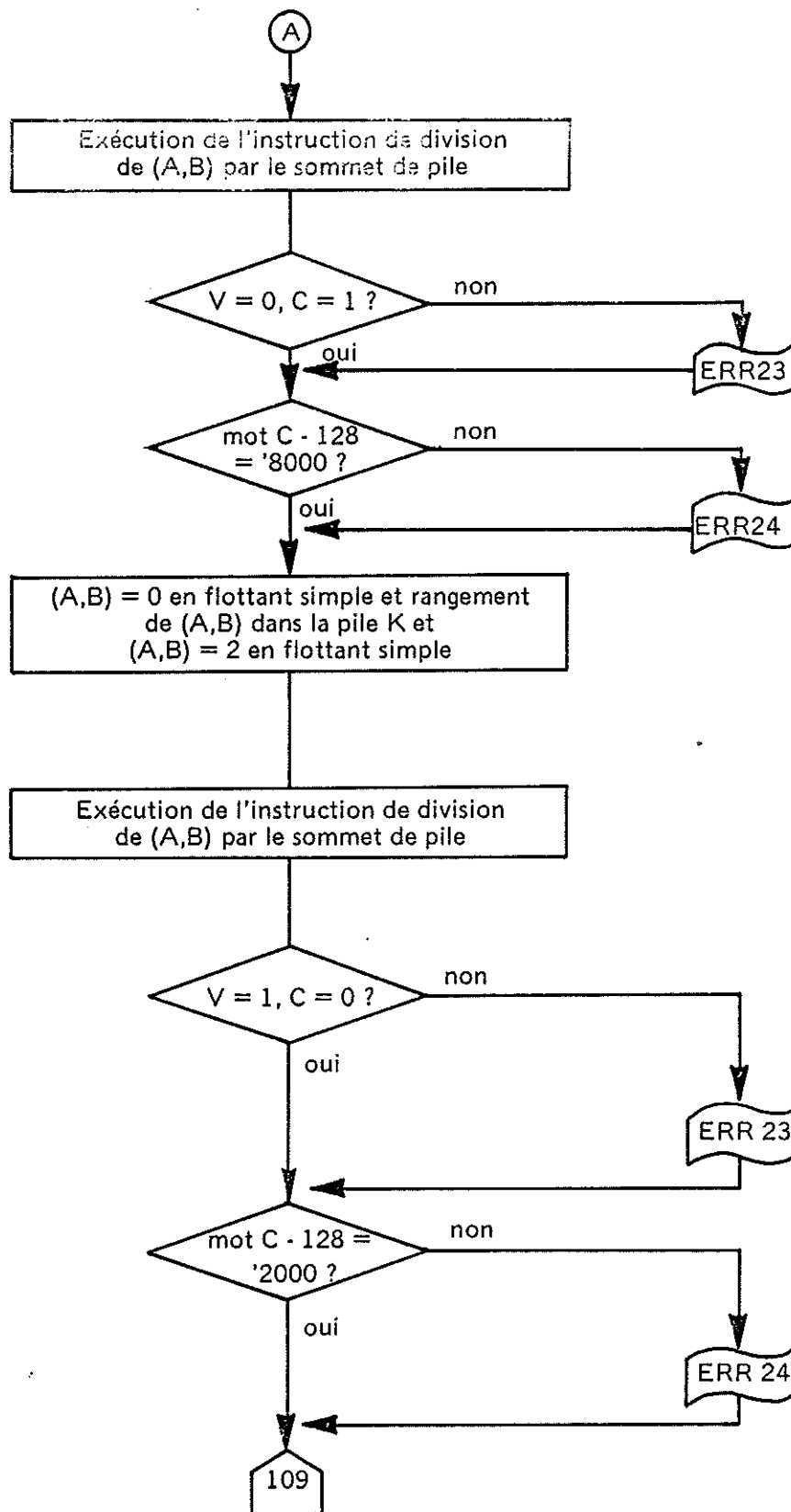
- Test division Accu, sommet de pile.

Le test vérifie que le résultat de l'opération est correcte. Pour cela, il effectue $(AB) / (\text{sommet de pile})$ avec $(AB) = 4$ et $(\text{sommet de pile}) = 2$ et vérifie que $(AB) = 2$.

D'autre part, il vérifie que le positionnement des indicateurs V et C dans ST et dans le mot C-128 est correct et que la division par 0 positionne bien le bit 2 du mot C-128.

Organigramme de la clé 109





CLE ACTION	110
Paramètre	non
Clé de PEC	oui
Clé de RNS	oui
<i>But</i> : Test multiplication accu, sommet de pile.	

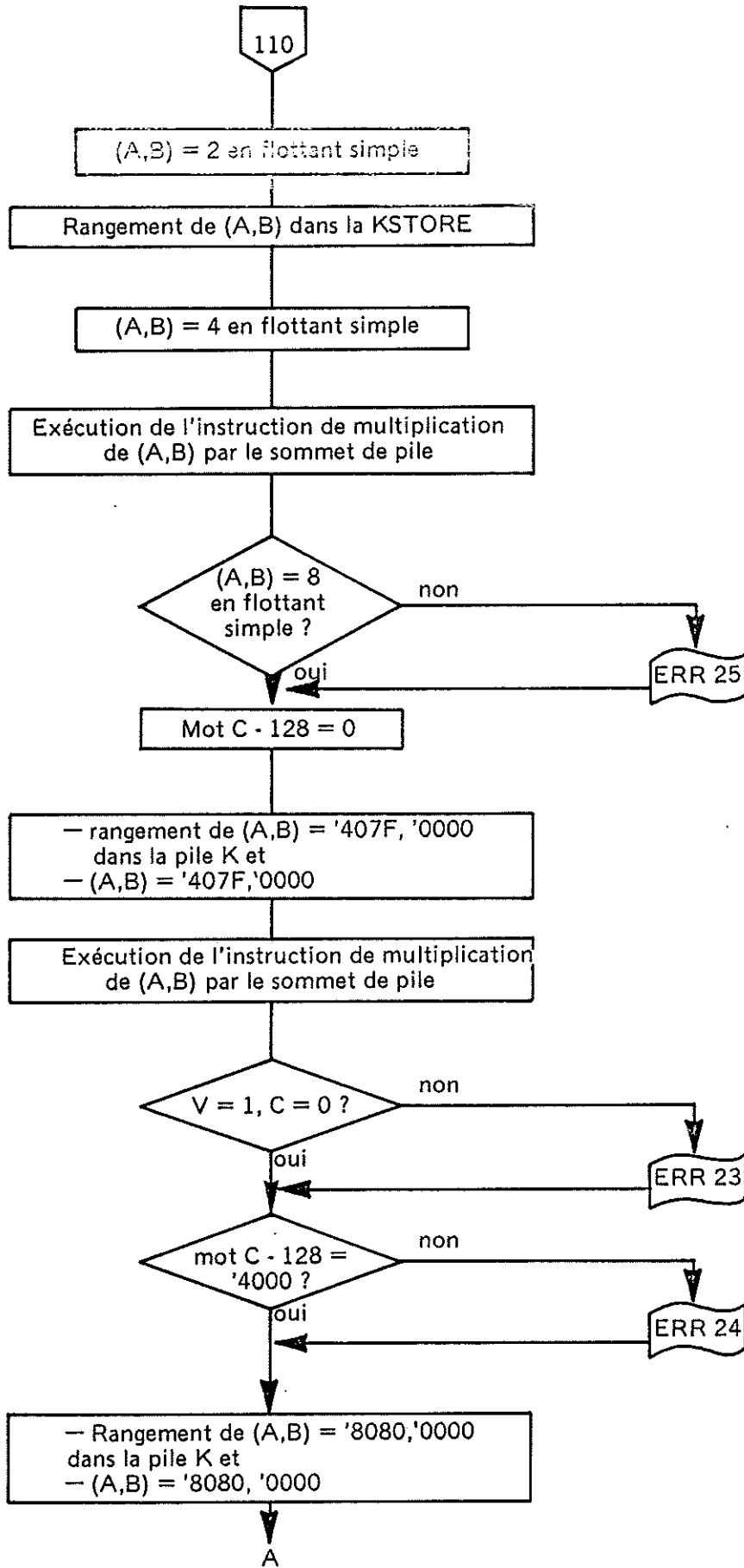
CLE 110 Test instruction flottant simple sur KSTORE.

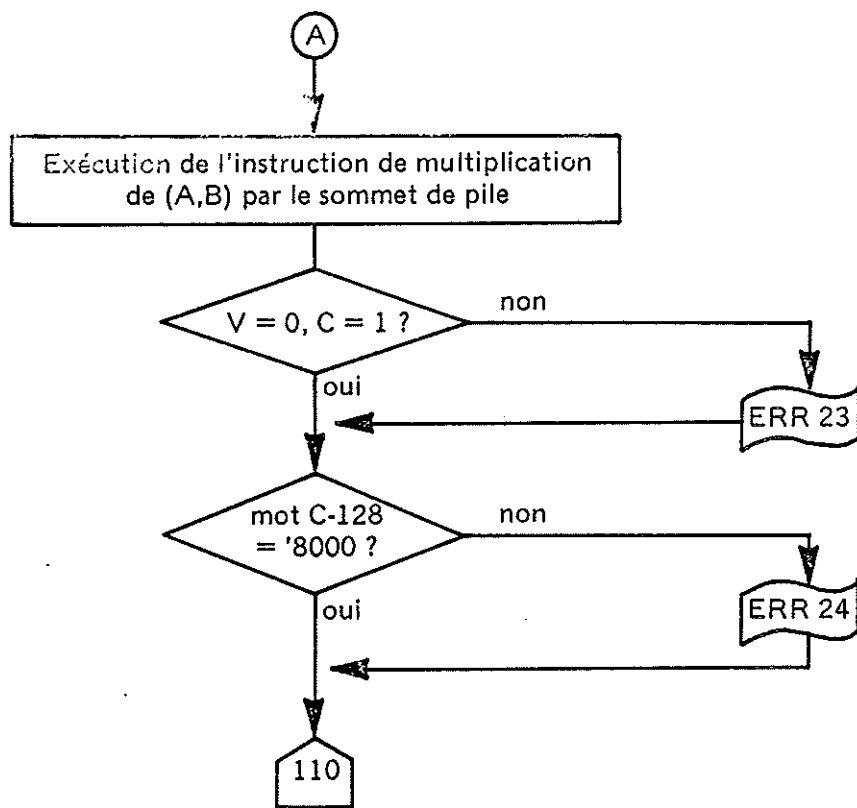
- Test multiplication accu, sommet de pile.

Le test vérifie que le résultat de l'opération est correcte.
Pour cela, il effectue $(AB) * (\text{sommet de pile})$ avec $(AB) = 4$
et $(\text{sommet de pile}) = 2$ et vérifie que $(AB) = 8$.

D'autre part, il vérifie que le positionnement des indicateurs
V et C dans ST et dans le mot C-128 est correct.

Organigramme de la clé 110





CLE ACTION	111
Paramètre	non
Clé de REC	oui
Clé de RNS	oui
But :	Test du schéma de Horner.

CLE 111 test du schéma de Horner (par l'intermédiaire de la fonction SINUS).

Vérification de $\text{SIN}(\pi)$; $\text{SIN}\left(\frac{\pi}{2}\right)$

- Le sinus est calculé par la fonction FORTRAN SIN, compilée avec option MI

- Les résultats attendus sont :

$\text{sin}(\pi) = '0080 \quad '0000$

$\text{sin}\left(\frac{\pi}{2}\right) = '7F00 \quad 'FFFF$

CLE ACTION	204
Paramètre	non
Clé de REC	oui
Clé de RNS	oui
But : Test indexation double.	

CLE 204 Cette clé effectue l'enchaînement
 104 0
 104 1
 104 2

CLE ACTION	205
Paramètre	non
Clé de REC	oui
Clé de RNS	oui
But : Test indexation triple.	

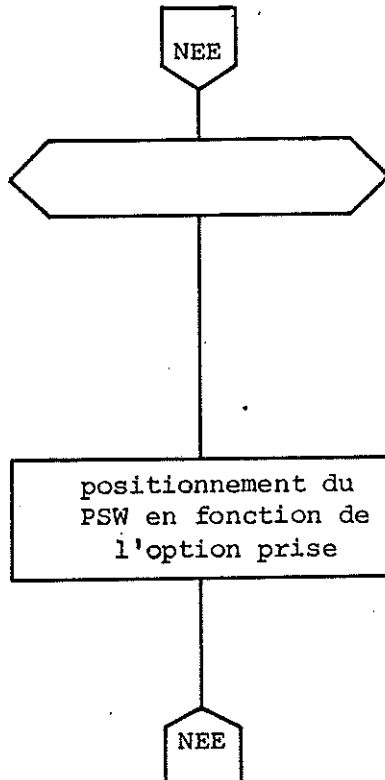
CLE 205 Cette clé effectue l'enchaînement
 105 0
 105 1
 105 2

3.2 - CLES OUTIL

Ce programme n'a que 2 clés outils.

CLE OUTIL	NEE
Paramètre	non
Clé de REC	non
Clé de RNS	non
<i>But : changer le niveau d'édition des messages d'erreurs.</i>	

Cette clé permet de changer le niveau d'édition des messages d'erreur. Cette option sera valable pour toutes les clés à venir même au-delà de la phase de test en cours, jusqu'à l'exécution d'une nouvelle clé NEE où jusqu'à une nouvelle définition du conversationnel.



Message :

"NIVEAU D'EDITION DES ERREURS?"

répondre par :

- 0 pour suppression des messages
- 1 pour messages niveau 1
- 2 pour messages niveau 1 et 2
- 3 pour messages niveau 1,2 et 3.

CLE OUTIL	EXC
Paramètre	1 = séquence fortran à tester
Clé de REC	non
Clé de RNS	non
<i>But</i> : Exécuter la séquence fortran compilé sans option MI et la fonction fortran compilée avec option MI.	

Le paramètre doit être pair, car il sert d'index dans la table des fonctions, or, pour chaque clé EXC, 2 fonctions sont exécutées.

Paramètre = 0 test du passage de paramètres
= 2 test du DO
= 4 test du GO TO calculé
= 6 test du GO TO au-delà de 128 mots
= 8 test indexation double - tableaux de réels
=10 test indexation double - tableaux d'entiers
=12 test indexation double - tableaux de flottant
double précision
=14 test indexation triple - tableaux de réels
=16 test indexation triple - tableaux d'entiers
=18 test indexation triple - tableaux de flottant
double précision.

3.3 - Tableau des clés disponibles au niveau 2

CLE	Paramètre	REC	RNS	Valeur du Paramètre	
100	non	oui	oui	/	test du passage de paramètres
101	non	oui	oui	/	test du DO avec pas égal à 1
102	non	oui	oui	/	test du GO TO calculé
103	non	oui	oui	/	test du GO TO au-delà de 128 mots
104	oui	non	non	0	test indexation double - tableaux réels
	1			1	test indexation double - tableaux entiers
				2	test indexation double - tableaux flottants double précision
105	oui	non	non	0	test indexation triple - tableaux réels
	1			1	test indexation triple - tableaux entiers
				2	test indexation triple - tableaux flottants double précision
106	non	oui	oui	/	test comparaison accu, sommet de pile
107	non	oui	oui	/	test addition accu, sommet de pile
108	non	oui	oui	/	test soustraction accu, sommet de pile
109	non	oui	oui	/	test division accu, sommet de pile
110	non	oui	oui	/	test multiplication accu, sommet de pile
111	non	oui	oui	/	test du schéma de Horner
204	non	oui	oui	/	test indexation double
205	non	oui	oui	/	test indexation triple.
REC	non				recette de bon fonctionnement
RNS	oui				recette de longue durée. Le paramètre indique le nombre de messages d'erreur tolérés.
END	non				sortie du programme de test.
PSW	non	non	non	/	mot d'état programme
STO	non	non	non	/	sauvegarde des clés
RST	non	non	non	/	restitution des clés
NEE	non	non	non	/	niveau d'édition des erreurs
EXC	oui 1	non	non	de 0 à 18	exécution d'une fonction fortran.

)

)

)

)