

4) net procédure ( argument )

SYNTAXE (PRIOR)  
 ( { [ ] [ variable chaîne index [ \*size ] ] nom } )

nomi décodé avec le type par défaut le dernier type au début par défaut type = variable

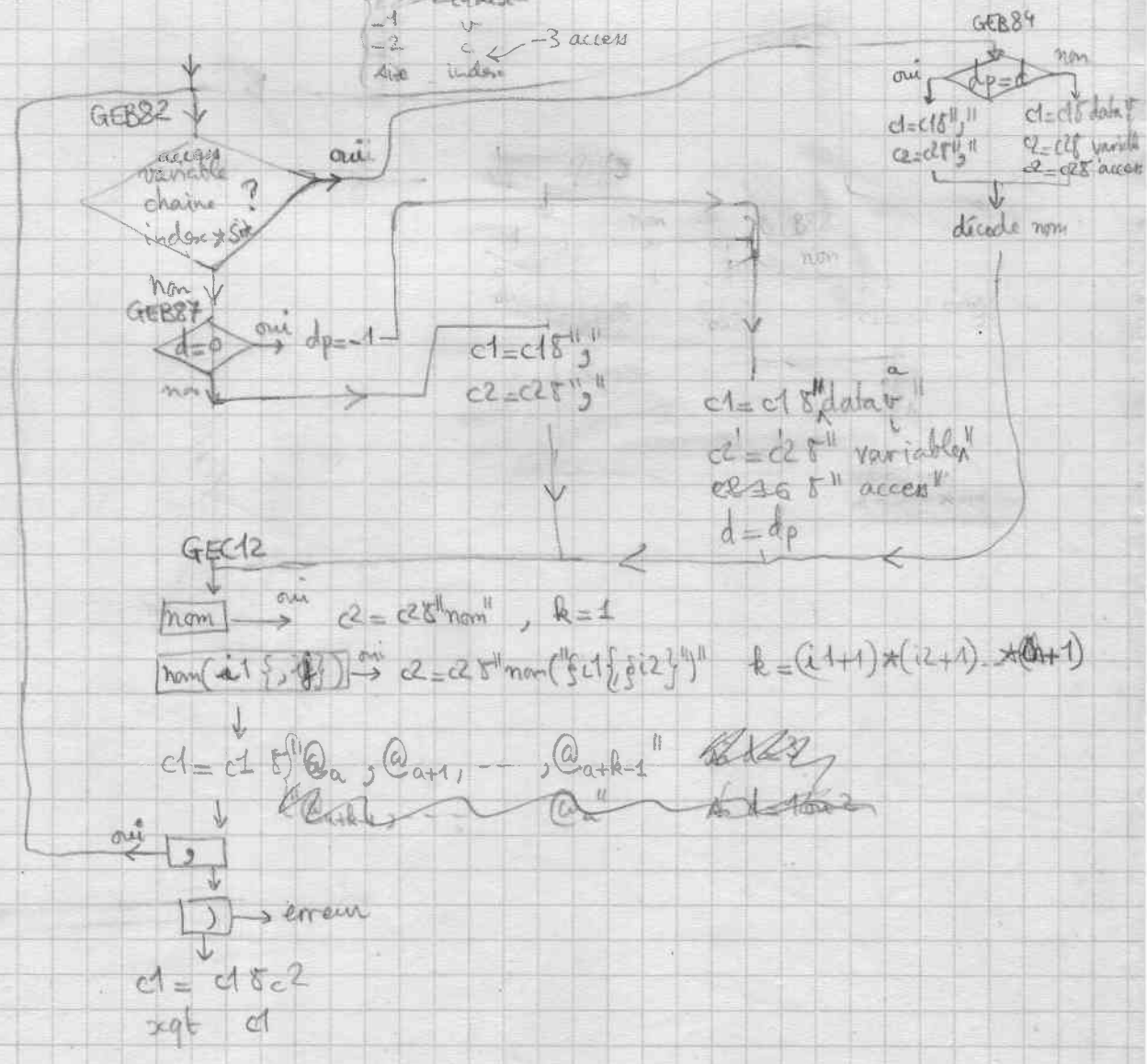
→ local data<sub>i</sub> <sup>deur deur</sup> @1, @2 ... @m @ data<sub>i</sub> @x @y | [ variable chaîne index n, {n} ] <sup>deur deur</sup> n {n}

nom(p, q)

ex eg

c1 = "local"  
 c2 = φ

d = 0 rien  
 -1 index  
 -2 v  
 -3 c  
 -4 index  
 a = 1 arg



④

procedure [ ( ) ]  
argument [ ( ) ]

GEB79:RTS

YPROC: CMP.B #"( ", (A5)  
BNE ~~ERR20~~ GEB79  
~~ERR20~~

WPROC: ADDQ #1, A5	← ADDQ #4, SP	↑ de le retour
MOVE.L # <del>0</del> <sup>global</sup> D3, D3		"local" (dans table des def)
BSR YPERCZI		peekc(D3) c <sub>1</sub> = "local"
BSR CPUSHNO		c <sub>2</sub> = ∅
LEA TBUF, A2		c <sub>2</sub>
LEA 1000(A2), A0		c <sub>1</sub>
MOVEQ #1, D3		arg = 1
MOVEQ #0, D2		d = 0

GEB82: MOVEM.L D2/D3/A0/A2, -(SP)

```

GEB82: BSR DECAN      decode nom
        BEQ  ERRIX
        TST  D0
        BMI  GEB87    -> nom inconnu

```

```

MOVEQ #-3, D3
CMP.L #gascon, A3
BEQ  GEB84

```

```

MOVEQ #-1, D3
CMP.L #gascon, A3
BEQ  GEB84

```

```

MOVEQ #-2, D3
CMP.L #gascon, A3
BEQ  GEB84

```

```

CMP.L #gascon+8, A3
BNE  GEB87

```

BSR SIZEIND

D0 = d  
D1 = arg  
A0 c1'  
A2 c2'  
D3 = dp

variable var.

chaîne

indose

nom connu mais pas type

```

GEB84: MOVEM.L (SP)+, D0/D1/A0/A2
        CMP  D0, D3
        BNE  GEB88

```

```

GEB84: MOVE D3, -(SP)
        BSR DECAN
        BEQ  ERRIX
        MOVE (SP)+, D3

```

saute au main suivant

```

GEB86: MOVEQ #",", D2
        MOVE.B D2, (A0)+
        MOVE.B D2, (A2)+
        BRA  GEC12

```

types inchangés

$c'_1 = c'_1 \& ", "$   
 $c'_2 = c'_2 \& ", "$

```

GEB88: LEA  GEB90, A3
        CMP #-1, D3
        BGT  GEB92
        SUBQ #8, A3
        BEQ  GEB92
        SUBQ #8, A3
        CMP #-3, D3
        BNE  GEB92
        SUBQ #8, A3

```

```

GEB87: MOVEM.L (SP)+, D3/D4/A0/A2

```

```

TST  D3
BNE  GEB86
MOVEQ #-1, D3

```

(variable] au début

⊗

D.L	gdaba-7	→ dataa	] } ⊗
D.L	gaccess-	→ access	
D.L	gdatab-7	→ datac	] }
D.L	get79i+1	→ chaine	
D.L	gdatab-7	→ data v	] }
D.L	get79v+1	→ variable	
GEB90:D.L	gdatabi-7	→ databi	
D.L	get79i+1	→ index	



```

GEB92: MOVEQ #32, D2
      MOVE.B D2, (A0)+
      MOVE.L (A3)+, A4

```

$$C'_1 = C'_1 \delta_{i, a} \text{data}$$

```

GEB94: MOVE.B (A4)+, (A0)+
      BNE GEB94
      MOVE.B D2, -1(A0)
      MOVE.B D2, (A2)+
      MOVE.L (A3)+, A4

```

$$C'_2 = C'_2 \delta_{i, \text{chain variable index} * \text{size access}}$$

```

GEB96: MOVE.B (A4)+, (A2)+
      BNE GEB96
      SUBQ #2, A2
      TST D3
      BMI GEC10
      MOVE.B #"*", (A2)+
      MOVE.B #"$", (A2)+
      MOVE D3, D0 ← MOVEQ #"0", D1
      ASR #4, D0
      BEQ GEB98
      ADD #1, D0
      MOVE.B D0, (A2)+

```

CMP D2, D3  
BEQ GEC10

→ chain variable index \* 32  
↓ index \* \$

\$20  
10  
2  
4  
2  
1

```

GEB98: MOVE D3, D0
      AND #$F, D0
      ADD #1, D0
      MOVE.B D0, (A2)+

```

```

GEC10: MOVE.B D2, (A2)+

```

(A)

GEC12: MOVE.B (A1)+, (A2)+

c2 = c2 8 "nom"

d3 = d  
d4 = arg  
A0  
A2  
d5 = k

c1  
c2

BNE GEC12

SUBQ #1, A2

MOVEQ #1, D5

{ MOVE D3, -(SP)  
MOVE.L D4, -(SP)

BSR DECCRPQ

(1)

BNE GEC16

→ non indicé

~~MOVE.B D1, (A2)+~~

(2) ou (1)

GEC14: MOVE.B D1, (A2)+

← concatène conserve D1/A5  
nouveaux A0 et A2

GEC14: MOVE.L D5, D1

BSR CONC12

BSR WADR

EXG A0, A2

MOVE.B #5, (A0)+

MOVE.L D3, D0

} valeur en décimal

BSR PINTA

← EXG A0, A2

ADDQ.L #1, D3

BSR LMUL13

MOVE.L D3, D5

BSR DECCRV6

(3)

BEQ GEC14

BSR DECCRPD

(1)

BNE ERRIX

MOVE.B D1, (A2)+

GEC16: MOVE.L (SP)+, D3

CMP #-3, (SP)

si d = -3

BNE GEC18

not d5 = 1

MOVEQ #1, D5

} 0

X

répète PB82+10v

GEC18: MOVE.L TMAPOCA, A3

MOVE -(A3), D4

BEQ ERRARN → autre que proc/funct

MOVE -(A3), D0

EXT.L D0

SUB.L D3, D0

BCS ERRARV

BSR PB840

LEA (A0, D0.L), A4

LEA TBUF+3680, A1

CMP.L A1, A4

BCC XERRM erreur minime

GEC20: MOVE.B (A3)+, (A0)+

DBRA D0, GEC20

SUBQ #1, A0

ADDQ.L #1, D3

LEA TBUF+1500, A1

CMP.L A1, A0

BGT GEC22 → concaténer

SUB #1000, A1

CMP.L A1, A2

BLT GEC23

GEC22: BSR CONC12

GEC23: SUBQ.L #1, D5

BEQ GEC25

MOVE.B #",", (A0)+

BRA GEC18

Conserve D3/D5/A5  
nouveaux A0/A2

GEC25: MOVE (SP)+, D2

BSR DECCRV ⑨

BEQ GEB82

BSR DECCRPD ⑩

BNE GEB82

BSR LB920

BRA YXQTI

← BSR CONC12  
terminé

concaténer c1 et c2 puis exécuter

d3 = arg0

d5 = nb arg

A0 c1  
A2 c2