


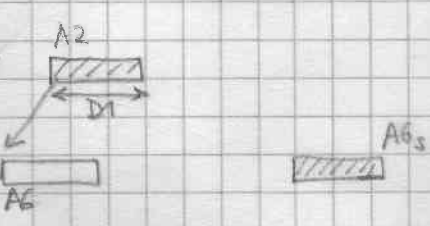
INOMI	③10	initialise table des noms
TSTA	③11	tests répartis //
INITS	③11a	vide : proc/for/exec/pile
INTR	12	initialise mémoire
NEXTAS	17	
DEC CH	20	décode chiffre en AS
DEC ...	20-21	// caractères spéciaux en AS
DEC N	22	décode et pose n
DECP	23	// $\left\{ \begin{matrix} p \\ q \end{matrix} \right\}$
<del>DECLT</del>	<del>24</del>	décode lettre
<del>DECPA</del>		// x-nom
		} remplaçable par utilisation de TYPCAR
DECAN	27	décode nom
XFLI1	29	écrit le nom (indexé) <sup>AS</sup> du litt/var/chaîne
XFLIT	31	// du litt D2 <sup>e</sup>
XIND1	32	détermine les indices pour XFLIT <sup>1</sup>
WNWTET <sup>(2)</sup>	32 <sup>e</sup>	inscrit tête du nom
WNWS	33	Inscrit le nom (non indexé) litt/index/var/chaîne/label
XNWL1	34	Transforme les labels \$SR → \$S1 (efface)
XNWLD	35	Efface les labels \$S1
XNWL	38a	Inscrit les labels de SOURCE1 et SOURCE / Réponse / error φ
XNWL1	36	// du bloc de ligne AS
XMARG4	38	met une marge de D.L + normal entre TMVNOM et TMVAKP
XMARG	38a	// normal
PACKC	39	pack si AS trop haut
PACK		// inconditionnel
ERRFAT	40	erreur fatale
XIVC	41	créé nouvelle variable / chaîne
XIVAR		= 0 = φ

32<sup>e</sup> inscrit tête de local

RERPOP	41a	clr si variable définie/sinon rétabli TVARND [boucle principale en cas d'erreur]
INITA	42	Initialise variables/labels avant run/debug/imm et écrit dr effectifs
INITRM	42a	"
WSDS LA88	43	Met var $A_0 = DO.W$ en type v met $\{A2\} = DO.L$ en libre
WPRIM	44	décode primary : crée $P_0$ ← LB16 (si crée $P_0 = \text{différent } d2$ )
LB303	48	$\{ = DO.W$
LB305	48	crée $P_0 = DO.L$
PUSHNZ	49	" $P_0 = 640$
PUSHNM		" $= 0 - 1$
PUSHNU		" $= 1$
PUSH13	49a	" $= P-1$ ← LB321 source $P_0 = U_{D2}$
PUSHN	51	" $P_0 = \{A_0\}$
APROCF	55	pour appel factia/proc
WNUMVL	56	met D2 = numéro du litt/var
WNUMI	58	met (A5, D2) pointeur sur index et type
WNUMIK	59a	" " " " " "
LMUL13	60	$DSL^S = D.L * DSL$ na signé (pour indice)
XIFAC XHFAC	61	convertir $\{P_{D_0}\}$ en type factorisé
XHPSE	62	recopie le contenu de $P_{D_0}$ en libre
XILOC	62a	Localise $P_0$ et $P_1$ : 
XIPOL	62b	Préparation $P_0 P_{-1}$
XHPOL	63	Préparation opération $P_0, P_1$ : conversions des types et copie en libre si type factorisé
NEWVP	64a	} inscrit nouvelle } inscript d'une nouvelle description de $P_{D_0}$ (poly)
PB30	65	
POP DO	65c	efface $P_{D_0}$
POP PR	65a	efface $P_{-1}$
POP NEW	65b	met $\{A2, A5\}$ dans $P_{-1}, D_0 P_0$
POP N	66	efface $P_0$
XHMUL	65	$P_{D_2} = P_{D_0} * P_{D_1}$
XHADD	67	$P_{D_2} = P_{D_0} + P_{D_1}$
XHSUB	68	$P_{D_2} = P_{D_0} - P_{D_1}$
XHDIV	69	$P_{D_2} = P_{D_0} / P_{D_1}$
XHDIVE	69a	" " division exacte si polynôme
XHCHA	71	$P_{D_2} = P_{D_0} \& P_{D_1}$ (chaîne)
XIMUL	66	$P_{-1} = P_0 * P_{-1}$ et efface $P_0$
XIADD	67	+
XISUB	68	-
XIDIV	69	/
XIEXP	69b,c	^
XICHA	70	& " (chaîne)

XCHAZE	69d	Rajoute ① à la fin de la chaîne P <sub>0</sub>
WPRIMS	72	décade ±primary et crée P <sub>0</sub>
WFACT	73	" fact "
WTERM	74	" terme "
XIPRIOR } XHPRIOR }	75	Convertir { P <sub>0</sub> P <sub>D0</sub> en priorité typev si possible
<del>WFLOAT</del>	76	décade flottant et crée P <sub>0</sub> $\{A_0\}$ pointe le flottant
XIFLO } XHFO }	77	Convertir { P <sub>0</sub> P <sub>D0</sub> en flottant
XIEXACT } XHEXACT } xhexactd	78	Convertir P <sub>0</sub> en exact de priorité typev P <sub>D0</sub> "
WEXPRS	79	décade exprs = {± terme}
WEXPRC	80	" exprs [comparais exprs] ou exprs dans comparais comparais
WDCOMP	81	" comparais
XIREEL } XHREEL }	82	Teste si { P <sub>0</sub> P <sub>D0</sub> est réel exact $\{A_0\}$ pointe P/q
XI POLY H	83	Teste si { P <sub>0</sub> P <sub>D0</sub> est poly var A <sub>2</sub>
XI ADR H	83	Teste si { P <sub>0</sub> P <sub>D0</sub> est adr oui ⇒ D3.L = valen
XIFLOA H	84	Teste si { P <sub>0</sub> P <sub>D0</sub> est flottant $\{A_2\}$
XIFLOB H	85	Teste si { P <sub>0</sub> P <sub>D0</sub> est réel exact ou flottant
WEXPR <sup>N</sup>	86	décade expr (avec opérations logiques) crée P <sub>0</sub>
WEXPRE <sup>O</sup>	86a	" exacte " WEXPRCOND 101 cond condit
WPOLY <sup>K</sup>	87	décade poly crée P <sub>0</sub> P <sub>A0</sub>
XHDEV <sup>H</sup>	87a	convertir P <sub>0</sub> en poly (erreur si non poly) test si P <sub>A0</sub> entier convertir P <sub>0</sub> en poly (si non poly met le numérateur)
WREEL	88	décade réel <sup>flottant ou</sup> exact, crée P <sub>0</sub> $\{A_0\} = 1/q$
WVGK		décade [k] défaut k=1 (index)
WINDEX <sup>V</sup>		décade k → D3.L = k
WVGK2	88a	décade [k] et met toppile à extrémité
WYGA	89	décade [a] défaut a=1 a=0
WVGAZ		
WADR <sup>V</sup>		décade a = adresse D3.L = a
WADRP		" a pair " = A0 = a
WENTIER <sup>V</sup>	89a	décade entier exact A0 pointe {p}
WENTIER <sup>V</sup>	90	" " ≠ 0, ±1 et met  p
		WADR 89.1 ) <sup>10</sup> accep WCINDEX " ) cent(x)

Code	Adresse	Description
<del>WMAT</del>	90	Décode nom matriciel
WEXCEND	93	dépile instruction de la pile exec
WINSTR	94	décode instruction en AS, traduit et exécute
	1 99	" sans traduire
WPILEASG	100a	décode [i] <sup>v</sup> et assigne P <sub>0</sub> (i = index ou variable)
WASGN	101	assignation : décode [=] et met la valeur de P <sub>D2</sub> ou index
WIFOND	101.1	effectue le calcul des conditions dans P <sub>0</sub> P <sub>D0</sub>
WASGN	102	Met P <sub>D0</sub> = P <sub>0</sub> et efface P <sub>0</sub>
ILB76	106a	copie bloc et vérif A6 {impair
PLB76		" pair, long 99...
LB76	107	" sans vérif pair, long paire
XLB76		"
LC75	108a	"
WCHAIN	109	décode vir chaîne
LB95A		met P <sub>0</sub> pour P <sub>D4</sub> (et D4 = numéro) et efface pack
LB95		LB95: met P <sub>0</sub> à gauche
PACK95	109a	
WCHAS	110	décode expr chaîne
CPUSHNO	111	crée P <sub>0</sub> = chaîne vide
		" 1 caractère D2.B
		D3L " D2.B
WCHAS1	112	décode chaînes
LC12	113	pour P <sub>0</sub> = $\begin{matrix} A2 \\ \text{     } \\ A6 \end{matrix}$ (A2 doit être au libu+6)
VSTR		
WNOMI	④	décode gauche d'une assignation
WNOMV	115	crée index (WNOMI) ou var (WNOMV) si incertain
WNOM		NOMTYPE m.détermine le type par terminaison
WNOMD		
WLABL	116	décode label
WMONO	117	décode monome unitaire crée P <sub>0</sub>
WVARK	118	décode v <sup>k</sup>
WVAR	118	décode v (littéral)
WRULE	119	décode v <sup>x</sup> y <sup>z</sup> = exprs et concatène au P <sub>0</sub> précédent
XHEXP	120	remplace P <sub>0</sub> par P <sub>0</sub> <sup>D3L</sup>
WSPBAS	121	exécution prog. basé de code 68000



WIDXP	123	décade [sexpr]) $d3 = \text{sexpr} (\text{entier} * 32)$
WIDX	124	décade $\begin{matrix} \min \\ \max \end{matrix}$ [sexpr])
WIDXI	125	d1.L = nb d'éléments du tableau A3/do d'index
WIDXV	126	" " de variable/chaine
WIDXF	127	vérif que d3 n'est pas en dehors du tableau
WSTVEL	130	décade réel et pointer élém d'ensemble
— C —	131	sexpr " " "
WPHPE	132	décade "[ ou ]"
WSTVSG	133	décade vel {x <sub>1</sub> ...x <sub>n</sub> } : ensemble réel
— C —	135	" cel {cel} : " de chaines
WSTVIN	136	" V-ensemble après (IN)
— C —	137	" C- " " "
XSTDF	138	pointer x-élé
XSTKH	139	pointer élément
XSTCF	140	compare p <sub>i</sub> et élém d <sub>i</sub> (chaines)
WTNAM	141	décade S(k <sub>1</sub> ...k <sub>n</sub> )
WTSRQ	142	" S(k <sub>1</sub> ...k <sub>n</sub> ), n, p
WTDES	143	" D(j <sub>1</sub> ...j <sub>n</sub> ), p'
WTPER	144	" P(l <sub>1</sub> ...l <sub>n</sub> )
XPEROK	145	vérifie que P(), N est une permutation de 1...n

} pour copy et sort

} pour sort