

LABEL 000000000PRINTER00175099CC EX OBJECT/READ,FILE SOURCEFILE=SYMBOL/RESEQB;END+0000

OBJECT /READ

SYMBOL/RESEQB

Data Document, Inc.

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

```

COMMENT: * TITLE: B5500/B5700 MARK XIV SYSTEM RELEASE * 00000100
* FILE ID: SYMBOL/RESEQB TAPE ID: SYMBOL2/FILE000 * 00000101
* THIS MATERIAL IS PROPRIETARY TO BURROUGHS CORPORATION * 00000102
* AND IS NOT TO BE REPRODUCED, USED, OR DISCLOSED * 00000103
* EXCEPT IN ACCORDANCE WITH PROGRAM LICENSE OR UPON * 00000104
* WRITTEN AUTHORIZATION OF THE PATENT DIVISION OF * 00000105
* BURROUGHS CORPORATION, DETROIT, MICHIGAN 48232 * 00000106
* * 00000107
* COPYRIGHT (C) 1971, 1972 BURROUGHS CORPORATION * 00000108
* AA320206 AA332366 AA386657 *; 00000109

```

```

BEGIN REAL COMMON; COMMENT CONTAINS VALUE POINTING TO 00100000
REAL ESP DISK. 00100100
00100200
00100300

```

```

COMMENT RR1, RR2, RR10 ARE USED IN DETERMINING WHETHER INPUT IS 00100500
VIA CARD OR DISK. 00100600

```

```

; REAL RR1, RR2, RR10; 00100700
00100800

```

```

FILE IN CARD DISK SERIAL (1,10,300); 00400200

```

```

FILE REMOTE 14(2,10); 00400210

```

```

SAVE FILE OUT KARD DISK SERIAL [20:600] (1,10,300,SAVE 1); 00400300

```

```

SAVE FILE OUT TBL DISK SERIAL [20:30] (2,30,300,SAVE 1); 00400400

```

```

FILE NEW DISK RANDOM(1,10,300,SAVE 1); 00400500

```

```

COMMENT ALL FORMAT STATEMENTS FOR RESEQB/TSHARER(ALMOST) 00500000

```

```

; 00500100

```

```

FORMAT SEQERR(19," SEQUENCE ERRORS."), 00500200

```

```

NOMATCH(" NO MATCHING SEQ. NU.",I10), 00500305

```

```

ERRHED(19," FIX-UP CARD GTR 72 CHR."); 00500400

```

```

COMMENT FORWARD DECLARATIONS FOR PROCEDURES APPEAR IN 00800000

```

```

SECTIONS 00800000 THRU 00900000 INCLUSIVE. 00800100

```

```

; 00800200

```

```

PROCEDURE STATEMENT; 00801000

```

```

FORWARD; 00801100

```

```

INTEGER PROCEDURE SCANNER; 00802000

```

```

FORWARD; 00802100

```

```

COMMENT DEFINES FOR CLASSES OF SYNTACTICAL ELEMENTS OF THE 00900000

```

```

LANGUAGE. 00900100

```

```

; 00900200

```

```

DEFINE STRINGID = 001#, 00900300

```

```

ARITHID = 002#, 00900400

```

```

STATSTART = 003#, 00900500

```

```

NUMBER = 004#, 00900600

```

```

STRING = 005#, 00900700

```

```

DEFFUNCID = 006#, 00900800

```

```

STDFUNCID = 007#, 00900900

```

```

STEPV = 008#, 00901000

```

```

TOV = 009#, 00901100

```

```

GOTOV = 010#, 00901200

```

```

COMMA = 011#, 00901300

```

```

RTPAREN = 012#, 00901400

```

```

LTPAREN = 013#, 00901500

```

```

TABV = 014#, 00901600

```

```

COLON = 015#, 00901700

```

```

SEMICOLON = 016#, 00901800

```

```

PERCENT = 017#, 00901900

```

```

THENV = 018#, 00902000

```

```

LBRACKET = 019#, 00902100

```

```

RBRACKET = 020#, 00902200

```

```

CROSHATCH = 021#, 00902300

```

```

FILEV = 022#, 00902400

```

	READV	= 023#,		00902500
	INPUTV	= 024#,		00902600
	PRINTV	= 025#,		00902700
1	OPTION	= 026#,		00902800
2	RELOP	= 027#,		00902900
3	ASGNOP	= 028#,		00903000
4	ADDOP	= 029#,		00903100
5	MULOP	= 030#,		00903200
6	EXPOPR	= 031#,		00903300
7	SSPART	= [42:6]#,		00903400
8	NEXTV	= 011#,		00903500
9	ENDV	= 004#,		00903600
10	CLASS	= [1:5]#,		00903610
11	INFOLINK	= [33:15]#,	%DEFINES FOR PARTIAL WORD SIZES	00903700
12	NUMBRART	= [6:27]#,	%AND OTHER THINGS.	00903800
13	RQWNDX	= [34: 6]#,		00903900
14	COLNDX	= [40: 8]#,		00904000
15	LINENO	= C#,		00904100
16	CURRENTLINENO	= CHARRY[0]#		00904200
17		;		00920000
18	COMMENT	GLOBAL DECLARATIONS FOR RESEQB/DISK ARE INCLUDED IN		01000000
19		SECTION 01000000.		01000100
20		;		01000200
21	COMMENT	SPECIAL IS AN ARRAY OF ELBAT WORDS FOR SPECIAL		01000300
22		CHARACTERS.		01000400
23		;		01000500
24	REAL ARRAY	SPECIAL[0:31];		01000600
25	COMMENT	RWARRY IS AN ARRAY CONTAINING THE "RESERVED" WORDS OF		01000700
26		BASIC AND CORRESPONDING ELBAT WDS.		01000800
27		;		01000900
28	REAL ARRAY	RWARRY[0:99];		01001000
29				01001100
30	INTEGER ARRAY	CTARRY[0:11];	COMMENT INDEX OF LAST ELEMENT IN EACH	01001200
31			GROUP.	01001300
32		;		01001305
33	SAVE ARRAY	OLD[0:9],	% ARRAYS USED BY SEQUENCER AND	01001310
34		LIN[0:9];	%CARDFIXUP TO PRODUCE NEW CARD IMAGE	01001320
35	DEFINE	SSFIRST = 0#,	% PARAMETERS USED BY SCANNER FOR	01001500
36		SSGROUPS = 6#,	% RECOGNIZING RESERVED WORDS.	01001600
37		IWFIRST = 42#,		01001700
38		IWGROUPS = 6#,		01001800
39		SS1 = 0#,		01001900
40		IW1 = 6#;		01002000
41	INTEGER	SCANMODE;	COMMENT 1-STATEMENT STARTER MODE,	01002100
42			2-IMBEDDED WORD MODE.	01002200
43			3-OPTION MODE,	01002300
44			;	01002400
45	DEFINE	SSMODE = 1#,		01002500
46		IWMODE = 2#,		01002600
47		OPMODE = 3#;		01002700
48	REAL ARRAY	CHARRY[0:77];	COMMENT CHARRY FILLED BY CARDCAN;	01002800
49	REAL ARRAY	CBLMN [0:77];	COMMENT COLMN[N] CONTAIN A NUMBER	01002900
50			POINTING TO THE CARD COLUMN IN	01003000
51			WHICH A CHARACTER IN THE CHARRY	01003100
52			WOULD HAVE APPEARED HAD NOT THE	01003200
53			BLANKS BEEN COMPRESSED OUT.	01003300
54			;	01003400
55	REAL	CNDX;	COMMENT INDEX OF LAST CHARACTER	01003500
56			SCANNED IN THE CHARRY.	01003600
57			;	01003700

INTEGER	NDX;		01003710
REAL	CC1,	% PTR. TO COL 1 OF CARD IMAGE	01003720
	CC73;	% PTR. TO COL 73 CARD IMAGE.	01003733
INTEGER	NBRISZ,	COMMENT LGTH OF NO.(DEC. FORM);	01003800
	NREC,	% RECORD # OF CURRENT CARD IMAGE.	01003810
	NBRCLM;	COMMENT STARTING POSITION OF NUMBER	01003900
		TO BE REPLACED. VALUES RETURNED	01004000
		BY SCANNER;	01004100
DEFINE	NEWLINENO = RR1#;	%NEWLINE NUMBER AS CALC. BY SEQUENCER	01004110
REAL	C;	COMMENT HOLDS VALUE OF NUMBER FROM	01004200
		SCANNER.	01004300
		;	01004400
BOOLEAN	EQUALMODETOG;	COMMENT TRUE = RELOP, FALSE = ASGNOP;	01004500
DEFINE	RELOPMODE = TRUE#;		01004600
	ASGNOPMODE = FALSE#;		01004700
DEFINE	EQUIV(EQUIV1,EQUIV2) =		01004710
	REAL(BOOLEAN(EQUIV1) EQV BOOLEAN(EQUIV2))		01004720
	= REAL(NOT FALSE)#;		01004730
COMMENT	WORDS USED BY SCANNER PASS TYPE QUANTITY SCANNED.		01004800
;			01004900
REAL	ELBAT;		01005000
INTEGER	ELCLASS;		01005100
COMMENT	LINETBL CONTAINS AN ENTRY FOR EACH LINE NUMBER REFERENCED		01005200
	BY A STATEMENT. ENTRIES HAVE THE FOLLOWING FORM.		01005300
	WD 1 [27:6] # OF BRANCH REFERENCES BY THIS STATEMENT.		01005400
	[33:15] RECORD # OF STATEMENT IN DISK FILE.		01005500
	WD 2 [1: 5] NUMBER OF COLUMNS(IN DEC) REQUIRED ON CARD		01005600
	BY OLD BRANCH REFERENCE.		01005700
	[6: 27] OLD LINE NUMBER,		01005800
	[40: 8] CARD COLUMN INDEX WHERE START OF NUMBER		01005900
	CAN BE FOUND ON OLD CARD.		01006000
	WD 3		01006100
	.	SAME AS WORD 2 FOR FURTHER OCCURENCES OF	01006200
	.	BRANCH LABELS.	01006300
	.		01006400
	WD N		01006500
	;		01006600
REAL ARRAY	LINETBL[0:63,0:255];		01006700
INTEGER	LINENDX,	COMMENT PTR TO LAST ENTRY IN LINETBL;	01006800
	LINEMAX,	COMMENT LASTUSED SLOT OF LINETBL;	01006810
	LINECTR;	COMMENT SET TO ZERO AT START OF	01006900
		STATEMENT SCAN;	01007000
DEFINE	NEXTLINWD=(LINENDX+LINENDX+1).ROWNDX,LINENDX.COLNDX#,		01007100
	LINEWD = LINENDX.ROWNDX,LINENDX.COLNDX#,		01007110
	LINEWD2 = (LINENDX+LINECTR+1).ROWNDX,(LINENDX+LINECTR+1)		01007120
	.COLNDX#,		01007130
	LINEWDN=(LINENDX+LINECTR).ROWNDX,(LINENDX+LINECTR).COLNDX#		01007200
	;		01007300
COMMENT	INFO CONTAINS A TWO WORD ENTRY FOR EACH DEFINED LINE		01007400
	NUMBER. THE FORM OF THE ENTRY IS:		01007500
	WD1 [6:27] OLD LINE NUMBER.		01007600
	[33:15] INFOLINK TO NEXT ENTRY. 0 IF LAST ENTRY.		01007700
	WD2 [6:27] NEW LINE NUMBER.		01007800
	;		01007900
REAL ARRAY	INFO[0:63,0:255];		01008000
COMMENT	SINFO SERVES AS A STACK HEAD.		01008100
	LINK = 0 IFF NO ENTRIES IN STACK		01008200
	18 DEC DIGIT LINE# CONVERTED TO ≤ 9 OCT DIG INTEGER TO		01008300
	GET STACKHEAD FROM STACK.		01008400
	;		01008500

	REAL ARRAY	SINFO[0:126];	01008600
	INTEGER	INFONDX; &INDEX INTO INFO ARRAY.	01008700
	DEFINE	INFOWD = INFONDX.ROWNDX,INFONDX.COLNDX#;	01008800
1		INFOWD2= (INFONDX+1).ROWNDX,(INFONDX+1).COLNDX#	01008900
2		;	01009000
3	INTEGER	NEXTAVAILINFO; & LAST USED INDEX INTO INFO.	01009100
4	DEFINE	NEXTINFO = (INFONDX+NEXTAVAILINFO+NEXTAVAILINFO+2).ROWNDX	01009200
5		,INFONDX.COLNDX#;	01009300
6	COMMENT	DISKIO OBTAINS A SEGMENT OF ESP DISK AT THE ADDRESS	02000000
7		SPECIFIED BY COMMON.	02000100
8		PARAMETERS IS ORDER OF APPERANCE ARE:	02000200
9		RW = 1 IF READ, = 0 IF WRITE.	02000300
10		ARRY=ARRAY DATA WILL BE READ INTO OR WRITTEN FROM.	02000400
11		N # OF WORDS TO BE READ OR WRITTEN.	02000500
12		DISKADR = DISK ADDRESS OF AREA TO BE WRITTEN OR READ	02000600
13		;	02000800
14	PROCEDURE	DISKIO(RW,ARRY,N,DISKADR);	02000900
15	VALUE	RW, N, DISKADR;	02001000
16	REAL ARRAY	ARRY[0];	02001100
17	REAL	RW,N, DISKADR;	02001200
18		COMMUNICATE(-8);	02001300
19	COMMENT	TBLHDR FIXES UP A HEADER FOR THE TBL FILE.	02200000
20		;	02200100
21	REAL STREAM PROCEDURE	TBLHDR(L); VALUE L;	02200200
22		BEGIN	02200300
23		DI + LOC TBLHDR; DS + 8 LIT "01T000 ";	02200400
24		DI + LOC TBLHDR; DI + DI+3;	02200500
25		SI + LOC L; DS + 3 DEC;	02200600
26		END OF TBLHDR;	02200700
27	COMMENT	OCTCNVT CONVERTS THE BCL NBR TO ITS OCTAL EQUIVALENT.	02300000
28		;	02300100
29	INTEGER STREAM PROCEDURE	OCTCNVT(NBR);	02300200
30	VALUE	NBR;	02300300
31		BEGIN	02300400
32		SI + LOC NBR; DI + LOC OCTCNVT; DS + 8 OCT;	02300500
33		END OF OCTCNVT;	02300600
34	COMMENT	DECCNVT CONVERTS THE OCTAL WD NBR TO ITS EQUIVALENT 8 CHR.	02350000
35		BCL REPRESENTATION.	02350100
36		;	02350200
37	REAL STREAM PROCEDURE	DECCNVT(NBR);	02350300
38	VALUE	NBR;	02350400
39		BEGIN	02350500
40		SI + LOC NBR; DI + LOC DECCNVT; DS + 8 DEC;	02350600
41		END OF DECCNVT;	02350700
42	COMMENT	GETABSADDR RETURNS THE ABSOLUTE ADDRESS OF THE PARAMETER	02360000
43		PASSED.	02360100
44		;	02360200
45	REAL STREAM PROCEDURE	GETABSADDR(L);	02360300
46		BEGIN	02360400
47		DI + L; GETABSADDR + DI;	02360500
48		END OF GETABSADDR;	02360600
49	COMMENT	CARDSCAN SCANS CARD COLUMN FOR <NUMERIC OR DIGIT >	03000000
50		CHARACTER OR < SPECIAL > CHARACTER. SEQUENCE NUMBER IN	03000100
51		COL 73-80 IS CONVERTED TO OCTAL AND PLACED IN ARRY[0].	03000200
52		;	03000300
53	STREAM PROCEDURE	CARDSCAN(CC1,CC73,ARRY);	03000400
54	VALUE	CC1,CC73;	03000500
55		BEGIN	03000600
56	LABEL	L1,L2;	03000700
57		SI + CC73; DI + ARRY; DS + 8 OCT; %CNVT LINE NO	03000800

	DI + CC73;	03000900
	3(DS + LIT "%"; DS + LIT ""); * INSERT STOPPER	03001000
	DI + ARRY; DI + DI+8; SI + CC1;	03001100
L1;	IF SC = ALPHA THEN	03001200
	BEGIN	03001300
	DS + 7 LIT "0";	03001400
L2;	DS + CHR; GO TO L1;	03001500
	END ;	03001600
	DS + 7 LIT "+000000";	03001700
	IF SC ≠ "%" THEN GO TO L2;	03001800
	DS + CHR;	03001900
	END CARDSCAN;	03003000
COMMENT	COMPRESSCARD, REMOVES BLANKS FROM THE CHARRY. A COMPANION	03100000
	ARRAY COLMN POINTS TO THE COLUMN WHERE THE CHARACTER	03100100
	APPEARED BEFORE COMPRESSION.	03100200
	;	03100300
PROCEDURE	COMPRESSCARD;	03100400
	BEGIN	03100500
INTEGER	TCNDX;	03100510
	CNDX + 0;	03100600
	NDX + 0;	03100700
	WHILE CHARRY[CNDX+CNDX+1] ≠ "%" DO	03100800
	BEGIN	03100900
	IF CHARRY[CNDX] ≠ "" THEN	03101000
	BEGIN	03101100
	CHARRY[NDX + NDX+1] + CHARRY[CNDX];	03101200
	COLMN[NDX] + CNDX;	03101300
	TCNDX + CNDX;	03101310
	END ;	03101400
	END ;	03101500
	CHARRY[NDX+NDX+1] + CHARRY[CNDX]; COLMN[NDX] + TCNDX+1;	03101600
COMMENT	END OF COMPRESSCARD;	03101700
	"STEP1" GETS THE NEXT ELBAT WORD AND STUFFS ELCLASS WITH	04200000
	THE CLASS OF QUANTITY SCANNED.	04200100
	;	04200200
INTEGER PROCEDURE	STEP1;	04200300
	STEP1 + ELCLASS + SCANNER;	04200400
COMMENT	"STEPIT" SAME AS STEP1 EXCEPT NO VALUE RETURNED.	04300000
	;	04300100
PROCEDURE	STEPIT;	04300200
	ELCLASS + SCANNER;	04300300
COMMENT	BLDID PACKS A CHARACTER INTO A WORD AT THE CHARACTER	04400000
	POSITION SPECIFIED BY CHRPOS.	04400100
	;	04400200
STREAM PROCEDURE	BLDID(WD, CHAR, CHRPOS);	04400300
VALUE	CHAR, CHRPOS;	04400400
	BEGIN	04400500
	DI + WD;	04400600
	DI + DI+CHRPOS;	04400700
	SI + LCC CHAR;	04400800
	SI + SI+7;	04400900
	DS + CHR;	04401000
	END OF BLDID;	04401100
COMMENT	SCANNER RETURNS AN ELBAT WORD AND ELCLASS WORD TO THE	04500000
	CALLER INDICATING THE TYPE QUANTITY SCANNED.	04500100
	;	04500200
INTEGER PROCEDURE	SCANNER;	04500300
	BEGIN	04500400
LABEL	COMPRESSCHAR, SKIPTOPERCENT;	04500500
LABEL	FRACTIONPART, DONOTHING;	04500600

	LABEL	PERIOD, LESSTHAN, ASTERISK, EQUAL, QUOTE,	04500700
		GREATERTHAN, TIMES, XIT;	04500800
	LABEL	LETTERONLY, LETTERDIGIT, ENDNUMBER;	04500900
1	INTEGER	NEXT; %TEMP FOR LOOK-AHEAD IN CHARRY	04501000
2	INTEGER	FSAVE, EXPON, CTNDX;	04501100
3	INTEGER	TCNDX, RWNDX;	04501200
4	INTEGER	CTR; %UTILITY COUNTER	04501300
5	REAL	T; %FOR GENERAL USE	04501400
6	SWITCH	SPECIALSW + PERIOD, LESSTHAN, ASTERISK, EQUAL, QUOTE,	04501500
7		GREATERTHAN, TIMES;	04501600
8	DEFINE	SPECIAL = [8:4]#; %IF NOT 0, SWITCH FR SPECIAL HNDLING	04501610
9	REAL	T1; %USED IN NUMBER BUILD	04501700
10	REAL	LOW, HIGH, %INDEX BRACKETS FOR CURRENT GROUP	04501800
11		MAX, %NUMBER OF GROUPS	04501900
12		ALPHAWD, %ACCUM FOR BUILDING RESERVED WD.	04502000
13		CTX ; %UTILITY COUNTER	04502100
14		IF T + CHARRY[CNDX + CNDX+1] < 0 THEN	04502200
15	BEGIN	COMMENT CHARACTER IS A SPECIAL CHARACTER;	04502300
16		IF T = "-" THEN T + "-";	04502400
17	COMPRESSCHAR:	T + T+2; %DECREASE INTERNAL CODE BY 2	04502500
18		ELBAT + SPECIAL[T,[45:3]&T[42:41:3]];	04502600
19		COMMENT NOTE COMPRESSION TECHNIQUE THAT DELETED THIRD BIT	04502700
20		FROM LEFT, AFTER DECREASING CODE BY 2.	04502800
21		;	04502900
22		IF T + ELBAT.SPECIAL = 0 THEN GO XIT;	04503000
23		GO TO SPECIALSWIT;	04503100
24	PERIOD:	C + 0; GO TO FRACTIONPART;	04503200
25	LESSTHAN:	IF T + CHARRY[T1 + CNDX+1] = ">" THEN	04503300
26	BEGIN		04503400
27		CNDX + T1;	04503500
28		T + "-x";	04503600
29		GO COMPRESSCHAR;	04503700
30	END;		04503800
31		IF T ≠ "=" THEN GO XIT;	04503900
32		CNDX + T1;	04504000
33		T + "≤";	04504100
34		GO TO COMPRESSCHAR;	04504200
35	ASTERISK:	IF CHARRY[T1 + CNDX+1] ≠ "*" THEN GO XIT;	04504300
36		CNDX + T1;	04504400
37		T + " "; %USE BLANK TO FAKE EXPON OPERATOR	04504500
38		GO COMPRESSCHAR;	04504600
39	EQUAL:	IF T + CHARRY[T1 + CNDX+1] = "<" THEN	04504700
40	BEGIN		04504800
41		CNDX + T1;	04504900
42		T + "≤";	04505000
43		GO COMPRESSCHAR;	04505100
44	END;		04505200
45		IF T = ">" THEN	04505300
46	BEGIN		04505400
47		CNDX + T1;	04505500
48		T + "≥";	04505600
49		GO COMPRESSCHAR;	04505700
50	END	;	04505800
51		IF EQUALMODETOG THEN GO XIT;	04505900
52		T + "+";	04506000
53		GO COMPRESSCHAR;	04506100
54	QUOTE:	WHILE(T+CHARRY[CNDX+CNDX+1]) ≠ "\"" DO	04506200
55		IF T = "\"" THEN GO XIT;	04506300
56		COMMENT NO MATCHING QUOTE, SO RETURN "x";	04506400
57		T + CHARRY[CNDX]; GO COMPRESSCHAR;	04506500

	GREATERTHAN: IF T + CHARRY[T1 + CNDX+1] = "<" THEN	04506600
	BEGIN	04506700
	CNDX + T1;	04506800
	T + "-"	04506900
	GO COMPRESSCHAR;	04507000
	END ;	04507100
	IF T ≠ "-" THEN GO XIT;	04507200
	CNDX + T1;	04507300
	T + "≥";	04507400
	GO COMPRESSCHAR;	04507500
	TIMES:	04507510
	IF T + CHARRY[T1+CNDX+2] & CHARRY[CNDX+1][36:42:6] =	04507600
	"EQ" THEN	04507700
	T + "="	04507800
	ELSE	04507900
	IF T = "NE" THEN	04508000
	T + "-"	04508100
	ELSE	04508200
	IF T = "GT" THEN	04508300
	T + ">"	04508400
	ELSE	04508500
	IF T = "LT" THEN	04508600
	T + "<"	04508700
	ELSE	04508800
	IF T = "GE" THEN	04508900
	T + "≥"	04509000
	ELSE	04509100
	IF T = "LE" THEN	04509200
	T + "≤"	04509300
	ELSE GO XIT; COMMENT IMPROPER RELATIONAL;	04509400
	CNDX + T1;	04509500
	GO COMPRESSCHAR;	04509600
	END OF SPECIAL CHARACTER HANDLING;	04509700
	IF T < 10 THEN	04509800
	BEGIN COMMENT NUMERIC DIGIT-ASSEMBLE NUMBER;	04509900
	TCNDX + CNDX;	04510000
	C + T;	04511000
	WHILE T + CHARRY[CNDX+CNDX+1] < 10 AND T ≥ 0 DO	04511100
	C + 10×C + T;	04511200
	NBRsiz + COLMN[CNDX]-COLMN[TCNDX]; NBRCLM + COLMN[TCNDX];	04512000
	IF SCANMODE = IWMODE THEN	04513000
	BEGIN	04514000
	IF T = "." THEN % SKIP OVER FRACTIONAL PART,	04514100
	FRACTIONPART: WHILE T + CHARRY[CNDX + CNDX+1] < 10 AND T ≥ 0 DO; %NOTHING	04514200
	IF T = "@" OR T = "E" THEN	04514300
	IF T + CHARRY[CNDX + CNDX+1] = "-" THEN	04514400
	DONOTHING: WHILE T + CHARRY[CNDX + CNDX+1] < 10 AND T ≥ 0 DO %NOTHING	04514500
	ELSE	04514600
	BEGIN	04514700
	IF T ≠ "+" THEN	04514800
	IF T < 0 OR T > 10 THEN CNDX + CNDX-1;	04514900
	GO DONOTHING;	04514910
	END ;	04515000
	END OF FRACTIONAL AND EXPONENT PARTS;	04515200
	ENDNUMBER:	04515300
	CNDX + CNDX-1;	04515400
	ELBAT + 0 & NUMBER[1:43:5];	04515500
	GO XIT;	04515600
	END OF NUMBER BUILDING	04515700
	ELSE COMMENT CHARACTER IS A LETTER-RECOGNITION PROCEDURE	04515800

	FOR RESERVED WORDS;	04515900
	IF NEXT ← CHARRY[CNDX+1] < 0 THEN	04516000
1	BEGIN	04516100
2	IF NEXT = "-" THEN	04516200
3	BEGIN COMMENT ELBAT WD FOR STRINGID;	04516300
4	ELBAT ← 0 & STRINGID[1:43:5];	04516400
5	CNDX ← CNDX+1;	04516500
6	GO XIT;	04516600
7	END ELSE	04516700
8	BEGIN COMMENT ELBAT WORD FOR ARITHID;	04516800
9	LETTERONLY:	04516900
10	LETTERDIGIT: ELBAT ← 0 & ARITHID[1:43:5];	04517000
11	GO XIT;	04517010
12	END ;	04517100
13	END ;	04517200
14	IF NEXT < 10 THEN	04517300
15	BEGIN COMMENT LETTER-DIGIT COMBINATION;	04517400
16	CNDX ← CNDX+1;	04517500
17	GO LETTERDIGIT;	04517600
18	END;	04517700
19	COMMENT RESERVED WORD RECOGNIZER;	04517800
20	TCNDX ← CNDX;	04517900
21	IF SCANMODE = SSMODE THEN	04518000
22	BEGIN COMMENT STATEMENT STARTER;	04518100
23	LOW ← SSFIRST; HIGH ← CTARRY[CTNDX+SS1];	04518200
24	MAX ← SSGROUPS;	04518300
25	END ELSE	04518400
26	IF SCANMODE = IWMODE THEN	04518600
27	BEGIN	04518605
28	LOW ← IWFIRST; HIGH ← CTARRY[CTNDX+IW1];	04518700
29	MAX ← IWGROUPS;	04518800
30	END ELSE GO SKIPTOPERCENT;	04518900
31	CTR ← 2;	04519000
32	CNDX ← CNDX + 1;	04519100
33	BLDID(ALPHAWD,T,1);	04519200
34	DO	04519300
35	BEGIN	04519400
36	BLDID(ALPHAWD,NEXT,CTR);	04519500
37	FOR RWNDX ← LOW STEP 2 UNTIL HIGH DO	04519600
38	IF EQUIV(ALPHAWD,RWARRY[RWNDX]) THEN	04519700
39	BEGIN COMMENT FOUND RESERVED WD., GET ELBAT WD;	04519800
40	IF (ELBAT+RWARRY[RWNDX+1]).CLASS = STDFUNCID THEN%DO04519900	
41	%NOTHING04520000	
42	ELSE	04520100
43	IF ELBAT.CLASS = DEFFUNCID THEN	04520200
44	IF NEXT ← CHARRY[CNDX+CNDX+1] < 10 THEN CNDX←CNDX-1;	04520300
45	GO XIT;	04520400
46	END OF GROUP OF N CHAR WDS;	04520500
47	LOW ← HIGH+2;	04520600
48	HIGH ← CTARRY[CTNDX+CTNDX+1];	04520700
49	END UNTIL	04520800
50	NEXT ← CHARRY[CNDX+CNDX+1] < 10 OR CTR ← CTR+1 ≥ MAX+1;	04520900
51	COMMENT NO MATCH, THEREFORE ASSUME LETTER IDENTIFIER;	04521000
52	CNDX ← TCNDX;	04521100
53	GO TO LETTERONLY;	04521200
54	SKIPTOPERCENT:	04521300
55	IF T ← CHARRY[CNDX] ≠ "-" THEN	04521305
56	WHILE T ← CHARRY[CNDX + CNDX+1] ≠ "-" DO;%NOTHING	04521400
57	GO COMPRESSCHAR;	04521500
	XIT:	04521600

```

SCANNER ← ELBAT.CLASS;                                04521700
END OF SCANNER;                                       04521800
COMMENT SCRAMLOOK SCRAMBLES A LINENO MOD 127 TO GET A STACK HEAD 05000000
FROM THE SINFO ARRAY THEN DOES A LINEAR SEARCH FOR A 05000100
MATCHING LINE NUMBER. SCRAMLOOK RETURNS TRUE IF FOUND, 05000200
ELSE FALSE. FORMAT PARAMETER LINENO WILL BE CONVERTED TO 05000300
FORM OF FIRST WORD OF INFO ENTRY WITH LINK PART POINTING 05000400
TO MATCHING ENTRY IN INFO IF FOUND, OR, IF NOT FOUND THEN 05000500
POINTING TO TOP OF STACK.                             05000600
;
BOOLEAN PROCEDURE SCRAMLOOK(LINENO);                   05000700
REAL LINENO;                                          05000800
BEGIN                                                05000900
LABEL FOUND;                                         05001000
INTEGER T;                                           05001100
IF (T ← INFONDX ← SINFO[LINENO MOD 127]) ≠ 0 THEN DO 05001200
IF INFO[INFOWD].NUMBPART = LINENO THEN                05001300
BEGIN                                                05001400
SCRAMLOOK ← TRUE;                                    05001500
GO FOUND;                                           05001600
END UNTIL INFONDX ← INFO[INFOWD].INFOLINK = 0;        05001700
INFONDX ← T;                                         05001800
SCRAMLOOK ← FALSE;                                   05001900
FOUND: LINENO ← 0 & LINENO[6:21:27] & INFONDX[33:33:15]; 05002000
END OF SCRAMLOOK;                                    05002100
COMMENT DEFLINENO MAKES A TOWORD ENTRY IN THE INFOARRAY FOR 05100000
DEFINED LINENO. SEE INFO DECLARATION FOR DESCRIPTION OF 05100100
THE ENTRY.                                           05100200
;
PROCEDURE DEFLINENO;                                  05100300
INTEGER BEGIN                                        05100400
T;                                                    05100700
C ← CURRENTLINENO;                                   05100900
T ← IF SCRAMLOOK(C) THEN C.INFOLINK                  05101100
ELSE 0;                                              05101200
IF (INFONDX ← T) = 0 THEN                             05101300
BEGIN                                                05101400
INFO[NEXTINFO] ← C;                                  05101500
SINFO[C.NUMBPART MOD 127] ← INFONDX;                 05101600
INFO[INFOWD2] ← 0 & NEWLINENO[6:21:27];              05101700
END;                                                 05101800
END OF DEFLINENO;                                    05102000
COMMENT ENTERLINETBL MAKES AN ENTRY IN LINETBL FOR EACH REF. 05102100
IN A STATEMENT TO A BRANCH NUMBER. SEE LINETBL DECLARA- 05500000
TION FOR FORM OF ENTRIES.                            05500100
;
PROCEDURE ENTERLINETBL;                               05500200
BEGIN                                                05500300
IF LINECTR = 0 THEN                                  05500400
LINETBL[NEXTLINEWD] ← NREC;                          05500600
LINECTR ← LINECTR + 1;                               05500700
LINETBL[LINENWDN] ← NBRCLM & NBRISZ[33:41:7]         05500800
& C[6:21:27];                                       05500900
END OF ENTERLINENUMBER;                              05600000
COMMENT COPY MOVES N CHARACTERS FROM THE OLD CARD BUFFER TO THE 05600100
NEW CARD BUFFER. ON EXIT, COPY POINTS TO THE NEXT AVAILABLE 05600200
CHARACTER IN THE NEW CARD BUFFER. FROMOLD CARD IS RESET TO 06000300
POINT AT THE LASTUSED CHARACTER + 1.                 06000400
;
REAL STREAM PROCEDURE COPY(N, FROMOLD CARD, TONNEW CARD); 06000500

```

```

VALUE      N, FROMOLDCARD, TONNEWCARD;          06000600
BEGIN                                           06000700
LOCAL     NDIV64, S;                            06000800
1         SI ← LOC N; DI ← LOC NDIV64;          06000900
2         SI ← SI+6; DI ← DI+7; DS ← CHR;       06001000
3         SI ← FROMOLDCARD; DI ← TONNEWCARD;    06001200
4         NDIV64(DS ← 32 CHR; DS ← 32 CHR); DS ← N CHR; 06001300
5         COPY ← DI;                             06001400
6         END OF COPY;                           06001600
7         COMMENT PUTLINE MOVES A LINE NUMBER OF N CHARACTERS INTO THE
8         POSITION SPECIFIED IN THE NEW CARD BUFFER, THE POINTER
9         FOR THE NEW CARD BUFFER IS UPDATED TO POINT AT THE NEXT
10        AVAILABLE CHARACTER.                   06100300
11        ;                                     06100400
12        REAL STREAM PROCEDURE PUTLINE(N, LINENO, CRDADDR); 06100500
13        VALUE      N, LINENO, CRDADDR;        06100600
14        BEGIN                                           06100700
15        SI ← LOC LINENO; DI ← CRDADDR;         06100800
16        N (DS ← CHR);                               06100900
17        PUTLINE ← DI;                               06101000
18        END OF PUTLINE;                             06101100
19        COMMENT ADVCRDADDR ADVANCES THE CHARACTER POINTER N CHARACTERS
20        ;                                     06200000
21        REAL STREAM PROCEDURE ADVCRDADDR(N, CRDADDR);    06200100
22        VALUE      N, CRDADDR;                  06200200
23        BEGIN                                           06200300
24        SI ← CRDADDR; SI ← SI+N;               06200400
25        ADVCRDADDR ← SI;                         06200500
26        END OF ADVCRDADDR;                       06200600
27        COMMENT GETSIGNUM COUNTS THE SIGNIFICANT DIGITS IN A DECIMAL NUMBER
28        AND PLACES THE NUMBER IN THE HIGH ORDER POSITION OF THE
29        WORD.                                       06300000
30        ;                                     06300100
31        INTEGER STREAM PROCEDURE GETSIGNUM(LINENO);    06300200
32        BEGIN                                           06300300
33        LOCAL     TEMP;                             06300400
34        SI ← LINENO; DI ← LOC TEMP; DS ← WDS;    06300500
35        SI ← LINENO; DI ← LOC TEMP;             06300600
36        7(IF SC = "0" THEN BEGIN DS ← LIT " "; SI ← SI+1; END ELSE
37        JUMP OUT);                                06300700
38        SI ← LOC TEMP; DI ← LINENO;             06300810
39        8(IF SC ≠ " " THEN BEGIN DS ← CHR; TALLY ← TALLY+1; END
40        ELSE SI ← SI+1);                          06300900
41        TEMP ← TALLY;                             06401000
42        SI ← LOC TEMP; DI ← LOC GETSIGNUM; DS ← WDS; 06401010
43        END OF GETSIGNUM;                         06401100
44        COMMENT COPY BLANKS COPIES N BLANKS TO SPECIFIED AREA. 06401200
45        ;                                     06401300
46        STREAM PROCEDURE COPYBLANKS(N, CRDADDR); VALUE N, CRDADDR; 06500000
47        BEGIN                                           06500100
48        LOCAL     T;                                 06500200
49        SI ← LOC N; DI ← LOC T; SI ← SI+6; DI ← DI+7; DS ← CHR; 06500300
50        DI ← CRDADDR; T(4(DS ← 16 LIT "
51        ; N(DS ← LIT " "));                       06500400
52        END OF COPY BLANKS;                       06500500
53        COMMENT CARDFIXUP SCANS LINETBL THEN USING SCRAMLOOK FINDS THE
54        NEW LINE NUMBER TO BE USED WHEN FIXING UP THE CARD IMAGE. 06500600
55        ;                                     06500700
56        PROCEDURE CARDFIXUP;                       07000000
57        BEGIN                                           07000100
                                           07000200
                                           07000300
                                           07000400

```

1	INTEGER	FROMCOL,UPTOCOL,	COMMENT COPY FROM COLUMN UP TO(BUT	07000500
		OLDNDX,NEWNDX,	NOT INCLUDING COLUMN;	07000600
			COMMENT POINTER TO NEXT AVAILABLE	07000700
			COLUMN;	07000800
2		INPTR, OUTPTR,	%POINTS TO NEXTUSED OR NEXTAVAIL AREA	07000900
3		NCHR,	% # OF SIGNIFICANT FIGS. IN LINE	07000905
4		RCDN,	%RECORD WHERE CARD IMAGE IS SAVED.	07000907
5		NFIX,	% # OF FIX-UPS FOR RECORD N.	07000908
6		INSIZ;	% # OF COLUMNS NBR RQD ON OLD CARD.	07000910
7	DEFINE	INCOL = UPTOCOL#;	% COLUMN WHERE # STARTED.	07000915
8	DEFINE	I = LINECTR#;	%INDEX	07000995
9	REAL	LINENO;	% ALMOST ALWAYS NEW LINE NO.(DEC)	07002000
10	LABEL	XIT,BACK,NEXTCRD;		07002500
11	COMMENT	ERROR PROCEDURE=CARDFIXUP.		07003000
12	;			07003100
13	PROCEDURE	ERR;		07003200
14	BEGIN			07003300
15		WRITE(REMOTE,ERRHED,DCICNVI(LIN[9]));		07003400
16	END	OF ERR;		07004000
17		IF LINENDX = 0 THEN GO XIT;	%NO FIX-UPS	07103000
18		LINEMAX + LINENDX; LINENDX + 0;		07103100
19		WHILE LINENDX < LINEMAX DO		07103105
20	BEGIN	%START OF LINETBL LOOP.		07103110
21		NEWNDX + 0; OLDNDX + 0;		07103120
22		RCDN + LINETBL[NEXTLINEWD],[33:15];		07103200
23		NFIX + LINETBL[LINWD],[27:6];		07103300
24		READ(NEW[RCDN],10,OLD[*]);		07103400
25		INPTR + RR10 + GETABSADDR(OLD);		07103500
26		OUTPTR + GETABSADDR(LIN);		07103600
27		COPYBLANKS(80,OUTPTR); LIN[9] + OLD [9];		07103605
28		FOR I + 1 STEP 1 UNTIL NFIX DO		07103700
29	BEGIN			07103800
30		IF I = 1 THEN		07103900
31	BEGIN			07104000
32		FROMCOL + 1;		07104100
33	END	;		07104200
34		UPTOCOL + LINETBL[LINWDN],[40:8];		07104210
35	BACK:	NCHR + UPTOCOL-FROMCOL;		07104300
36		IF(NEWNDX+NEWNDX+NCHR)>72 THEN BEGIN ERR; GO NEXTCRD;END;		07104400
37		OLDNDX + OLDNDX + NCHR;		07104500
38		OUTPTR + COPY(NCHR,INPTR,OUTPTR);		07104600
39		INSIZ + LINETBL[LINWDN],[33:7];		07104800
40		INCOL + LINETBL[LINWDN],[40:8];		07104900
41		IF SCRAMLOOK(LINENO + LINETBL[LINWDN].NUMBPART) THEN		07105000
42	BEGIN			07105005
43		IF INFO[INFOWD],[1:5] = 0 THEN		07105010
44	BEGIN			07105100
45		LINENO + DECCNVT(INFO[INFOWD2].NUMBPART);		07105110
46		INFO[INFOWD],[1:5] + NCHR + GETSIGNUM(LINENO);		07105120
47		INFO[INFOWD2] + LINENO;		07105125
48	END	ELSE		07105130
49	BEGIN			07105200
50		LINENO + INFO[INFOWD2];		07105210
51		NCHR + INFO[INFOWD],[1:5];		07105300
52	END	;		07105400
53	END	ELSE		07105600
54	BEGIN			07105700
55		WRITE(REMOTE,NOMATCH,LINETBL[LINWDN]);		07105710
56		LINENO + DECCNVT(LINENO,NUMBPART);		07105720
57		NCHR + GETSIGNUM(LINENO);		07105730

```

END ;                                07105740
IF(NEWNDX+NEWNDX+NCHR)>72 THEN BEGIN ERR; GO NEXTCRD; END; 07106000
OUTPTR + PUTLINE(NCHR,LINENO,OUTPTR); 07106100
OLDNDX + INCOL + INSIZ;              07106200
RR1 + OLDNDX-1;                      07106290
INPTR + ADVCRDADDR(RR1,RR10);        07106300
IF I = NFIX THEN                     07106400
BEGIN                                 07106500
  IF NCHR + LINETBL[LINENO],[20:71] ≠ 0 THEN 07106600
  BEGIN                               07106610
    NCHR + NCHR-OLDNDX;              07106620
    IF (NEWNDX + NEWNDX+NCHR) > 72 THEN 07106630
    BEGIN                             07106640
      ERR; GO NEXTCRD;              07106650
    END ;                             07106660
    OUTPTR + COPY(NCHR,INPTR,OUTPTR); 07106670
  END ;                               07106700
  END ELSE                           07106800
  BEGIN                               07106900
    FROMCOL + OLDNDX;              07107000
  END ;                               07107200
  END ;                               07107400
NEXTCRD: WRITE(NEW[RCON],10,LIN[*]); 07107500
LINENDX + LINENDX+NFIX;             07107600
END OF LINETBL LOOP;               07107700
XIT: END OF CARDFIXUP;             07108000
COMMENT PROCEDURES IN SECTION 08000000 ARE CALLED BY THE STATEMENT 08000000
ROUTINE. THE STATEMENT IS SCANNED FOR THE PRESENCE OF A 08000100
LINE NUMBER(S). IF FOUND, THE NUMBER IS ENTERED IN 08000200
LINETBL. 08000300
; COMMENT IFST PROCESSES IF STATEMENTS. 08000400
COMMENT 08100000
; 08100100
PROCEDURE IFST; 08100200
LABEL BEGIN 08100300
  XIT; 08100305
  EQUALMODETOG + RELOPMODE; 08100400
  WHILE STEPI ≠ THENV DO 08100500
    IF ELCLASS = PERCENT THEN GO XIT; 08100600
  SCANMODE + SSMODE; 08100690
  IF STEPI = NUMBER THEN 08100700
  ENTERLINETBL 08100800
  ELSE 08100900
  IF FLCLASS = STATSTART THEN 08101000
  BEGIN 08101100
    EQUALMODETOG + ASGNOPMODE; 08101200
    STATEMENT; 08101300
  END; 08101400
XIT: END OF IFST; 08101500
COMMENT "ON" STATEMENT PROCESSOR. 08200000
; 08200100
PROCEDURE ONST; 08200200
LABEL BEGIN 08200300
  XIT, BACK; 08200400
  WHILE STEPI ≠ GOTOV DO 08200500
    IF ELCLASS = PERCENT THEN GO XIT; 08200600
  BACK: IF STEPI = NUMBER THEN 08200700
  ENTERLINETBL ELSE GO XIT; 08200800
  IF STEPI = COMMA THEN GO BACK; 08200900
XIT: END OF ONST; 08201000

```

COMMENT	"EOF" LABEL PROCESSOR. SEARCHES FOR EOF LABELS	08300000
;		08300100
PROCEDURE	EOFST;	08300200
1 BEGIN		08300300
2 LABEL	XIT;	08300400
3	WHILE STEPI ≠ PERCENT DO	08300500
4	IF ELCLASS = COLON THEN	08300600
5 BEGIN		08300700
6	IF STEPI = NUMBER THEN	08300800
7 BEGIN		08300805
8	ENTERLINETBL;	08300810
9	WHILE STEPI ≠ PERCENT DO; %NOTHING	08300815
10	LINETBL[(LINENW)].[20:7] + COLMN[CNEX];	08300820
11	END;	08300825
12	GO XIT;	08300900
13	END ;	08301000
14	XIT; END OF EOFST;	08301100
15	COMMENT "MATRIX" STATEMENT PROCESSOR.	08400000
16	;	08400010
17	PROCEDURE MATST;	08400100
18	BEGIN	08400200
19	IF STEPI = INPUTV OR ELCLASS = PRINTV THEN	08400300
20	EOFST;	08400400
21	END OF MATST;	08400500
22	COMMENT "GOSUB" STATEMENT PROCESSOR.	08500000
23	ALSO USED BY "GO TO".	08500010
24	;	08500100
25	PROCEDURE GOSUBST;	08500200
26	BEGIN	08500300
27	IF STEPI = NUMBER THEN ENTERLINETBL;	08500400
28	END OF GOSUBST;	08500500
29	PROCEDURE STATEMENT;	08900000
30	BEGIN	08900100
31	LABEL EXIT;	08900200
32	SCANMODE + IWMODE;	08900400
33	CASE ELBAT,SSPART OF	08900500
34	BEGIN	08900600
35	IFST;	08900700
36	ONST;	08900800
37	COMMENT DEFST; GO EXIT;	08900900
38	COMMENT DIMST; GO EXIT;	08901000
39	COMMENT ENDST; GO EXIT;	08901100
40	COMMENT FORST; GO EXIT;	08901200
41	COMMENT LETST; GO EXIT;	08901300
42	MATST;	08901400
43	COMMENT REMST; GO EXIT;	08901500
44	COMMENT DATAST; GO EXIT;	08901600
45	COMMENT GOTOST; GOSUBST;	08901700
46	COMMENT GO EXIT; GO EXIT; % NEXT STATEMENT	08901800
47	COMMENT READST; GO EXIT;	08901900
48	COMMENT STOPST; GO EXIT;	08902000
49	COMMENT FILEST; GO EXIT;	08902100
50	GOSUBST;	08902200
51	COMMENT INPUTST; EOFST;	08902300
52	COMMENT PRINTST; EOFST;	08902400
53	COMMENT WRITEST; GO EXIT;	08902500
54	COMMENT RETURNST; GO EXIT;	08902600
55	COMMENT RESTOREST; GO EXIT;	08902700
56	END OF CASE STATEMENT;	08902800
57	EXIT;	08902900

```

END OF STATEMENT;                                08903000
COMMENT SEQUENCER IS THE DIRECTOR OF THE WHOLE AFFAIR. IT CALLS THE 09000000
ROUTINES NECESSARY TO PRODUCE THE RESEQUENCED FILE. 09000100
;                                                    09000200
PROCEDURE SEQUENCER;                                09000300
BEGIN                                                09000400
LABEL START,EOF;                                    09000500
INTEGER I,REC,BASE,INCR,FROM,TOO,LLN,LSEQ,          09000510
LASTLINENO, LASTSEQ;                                09000515
DEFINE SEQ = [21:27]#;                               09000525
BOOLEAN RESEQ,TBLOUT;                                09000527
SAVE ARRAY ESP[0:29];                                09000530
REAL ERRCNT;                                         09000540
LASTLINENO + LSEQ + -(LASTSEQ + 100000000); %***** 09000600
ESP[0] + 0;                                          09000700
DISKIO(1,ESP,30,COMMON);                             09000800
TBLOUT + ESP[9],[6:12] = "1S";                       09001000
IF BASE + ESP[7] > 0 THEN                             09001100
BEGIN                                                09001200
RESEQ + TRUE; INCR + ESP[8].SEQ;                      09001300
FROM + ESP[5]; TOO + ESP[6].SEQ;                     09001400
END ;                                                09001500
RR1 + TBLHDR((LLN+ESP[1],[33:15]),[40:8]);           09001600
IF TBLOUT THEN FILL TBL WITH RR1,ESP[2];             09001700
FILL CARD WITH ESP[3],ESP[4];                        09001800
FILL KARD WITH ESP[9],ESP[2];                        09001900
FILL NEW WITH ESP[9],ESP[2];                         09001910
START:                                               09002000
READ(CARD,10,LIN[*]) [EOF];                          09002010
CC73 + GETABSADDR(LIN[9]);                            09002020
CC1 + CC73-9;                                         09002030
CARDSCAN(CC1,CC73,CHARRY);                          09002040
COMPRESSCARD;                                        09002050
IF CURRENTLINENO ≤ LASTLINENO THEN                  09002100
ERRCNT + ERRCNT+1                                   09002105
ELSE LASTLINENO + CURRENTLINENO;                    09002200
IF RESEQ THEN                                        09002300
IF CURRENTLINENO ≥ FROM AND CURRENTLINENO ≤ TOO THEN 09002400
BEGIN                                                09002500
LIN[9] + DECCNVT(BASE);                              09002600
BASE + (RR1+BASE)+INCR;                              09002700
END ELSE LIN[9] + DECCNVT(RR1+CURRENTLINENO);        09002800
IF RR1 ≤ LSEQ THEN % RR1 IS NEW LINE NUMBER.         09002900
BEGIN                                                09003000
RR1 + LSEQ+2; LIN[9] + DECCNVT(RR1);                 09003100
END ;                                                09003200
COMMENT WRITE "CARD" IN NEW FILE. MAY FIXUP LSTER;  09003300
WRITE(KARD,10,LIN[*]);                               09003500
LINECTR + 0;                                         09003600
DEFLINENO; % MAKE ENTRY FOR LINE# IN INFO ARRY.     09003700
EQUALMODETOG + ASGNOPMODE;                          09003800
SCANMODE + SSMODE;                                  09003900
CNDX + 0;                                            09003990
IF STEPI = STATSTART THEN STATEMENT;                09004000
IF LINECTR ≠ 0 THEN                                  09004100
BEGIN                                                09004200
LINETBL[LINENWD],[27:6] + LINECTR;                  09004300
LINENBX + LINENDX + LINECTR; LINECTR + 0             09004400
END;                                                 09004500
LSEQ + RR1; NREC + NREC+1;                           09004600

```

Data Documents/Inc.

```

IF TBL OUT THEN          COMMENT BUILD TABLE FILE;          09004700
BEGIN                    *ESP IS TEMPORARY STORAGE NOW.      09004800
  ESP[(REC+1) MOD 30] := NEWLINEND & REC[4:32:16];          09004900
  REC = REC + 1;                                             09005100
  IF I = 29 THEN                                             09005200
    WRITE(TBL,30,ESP[*]);                                     09005300
  END ;                                                       09005400
  GO START;                                                  09005500
  IF TBL OUT THEN      * WRITE OUT TABLES                    09005600
  BEGIN                                                       09005700
    ESP[(REC+1) MOD 30] := LASTSEQ;                          09005800
    WRITE(TBL,30,ESP[*]);                                     09005900
    READ(TBL[0],30,ESP[*]);                                   09006000
    ESP[0] := REC;                                           09006100
    WRITE(TBL[0],30,ESP[*]);                                  09006200
    LOCK(TBL,*);                                             09006300
  END ;                                                       09006400
  CLOSE(CARD);                                               09006500
  LOCK(KARD,*);                                              09006510
  COMMON = LSEQ;                                             09006600
  IF ERRCNT > 0 THEN WRITE(REMOTE,SEQERR,ERRCNT);           09006610
  CARDFIXUP;                                                 09006700
  END OF SEQUENCER;                                          09010000
  COMMENT INITIALIZATION FOR RESEQB/DISK.                    10000000
  ;
  FILL SPECIAL[*] WITH                                       10000100
  OCT2500000000000000, *CROSSHATCH                          10000200
  OCT13, *COMMERCIAL AT                                       10000300
  OCT14, *UNASSIGNED(QUESTION MARK)                          10000400
  OCT1700000000000000, * COLON                               10000500
  OCT1700000000000000, * COLON                               10000600
  OCT3306706341250225, * >                                   10000700
  OCT3300716242250125, * ≥                                   10000800
  OCT35000000000000101, * + ADD OPERATOR                     10000900
  OCT21, * A UNASSIGNED                                       10001000
  OCT3201000000000000, * . UNASSIGNED, ARITHMETIC DEC. PT. 10001100
  OCT2300000000000000, * [                                   10001200
  OCT34, * & UNASSIGNED                                       10001300
  OCT1500000000000000, * (                                   10001400
  OCT3302627101254225, * <                                   10001500
  OCT3400000000000000, * ←                                   10001600
  OCT00070000000000040, * x UNASSIGNED, BEG OF RELATIONAL 10001700
  OCT41, * J UNASSIGNED,                                       10001800
  OCT52, * $ UNASSIGNED,                                       10001900
  OCT36030000000000401, * * MULOP, "***=EXPONENT OPERATOR. 10002000
  OCT35000000000000301, * - ADDOP, MINUS                     10002100
  OCT1400000000000000, * )                                   10002200
  OCT2000000000000000, * ; SEMI-COLON,                       10002300
  OCT3300637002254125, * ≤                                   10002400
  OCT3700000000000000, * EXPONENT OPERATOR                   10002500
  OCT36000000000001001, * / DIV OPERATOR                     10002600
  OCT1300000000000000, * , CUMMA                             10002700
  OCT2100000000000000, * % PERCENT                           10002800
  OCT3300606144250425, * ×                                   10002900
  OCT3304616004254425, * =                                   10003000
  OCT2400000000000000, * J                                   10003100
  OCT05050000000000077, * " UNASSIGNED, BEGIN STR          10003200
  OCT0, * 0 UNASSIGNED                                        10003300
  OCT1; * 1 UNASSIGNED                                        10003400
  FILL RWARRY[*] WITH *NDX                                   10003500
  "0IF0000", * 0 STATEMENT STARTERS.                         10003600

```

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

Data Documents/Inc.

	OCT0300000000000000,		10003700
	"00N000C0",	% 2	10003800
	OCT0300C00000000001,		10003900
1	"0DEF0000",	% 4	10004000
2	OCT0300C00000000002,		10004100
3	"0DIM0000",	% 6	10004200
4	OCT0300C00000000003,		10004300
5	"0END00C0",	% 8	10004400
6	OCT0300000000000004,		10004500
7	"0FOR0000",	% 10	10004600
8	OCT0300000000000005,		10004700
9	"0LET0000",	% 12	10004800
10	OCT0300C00000000006,		10004900
11	"0MAT0000",	% 14	10005000
12	OCT0300C00000000007,		10005100
13	"0REM0000",	% 16	10005200
14	OCT0300000000000010,		10005300
15	"0DATAC00",	% 18	10005400
16	OCT0300000000000011,		10005500
17	"0GOT0000",	% 20	10005600
18	OCT0300C00000000012,		10005700
19	"0NEXT000",	% 22	10005800
20	OCT0300C00000000013,		10005900
21	"0READ000",	% 24	10006000
22	OCT0300C00000000014,		10006100
23	"0STOPC00",	% 26	10006200
24	OCT0300000000000015,		10006300
25	"0FILES00",	% 28	10006400
26	OCT0300000000000016,		10006500
27	"0GDSUB00",	% 30	10006600
28	OCT0300000000000017,		10006700
29	"0INPUT00",	% 32	10006800
30	OCT0300000000000020,		10006900
31	"0PRINT00",	% 34	10007000
32	OCT0300C00000000021,		10007100
33	"0WRITE00",	% 36	10007200
34	OCT0300C00000000022,		10007300
35	"0RETURN0",	% 38	10007400
36	OCT0300000000000023,		10007500
37	"0RESTORE",	% 40	10007600
38	OCT0300000000000024,		10007700
39	"0FN00000",	% 42 DEF FN ID	10007800
40	OCT0600000000000000,		10007900
41	"0TD0000C0",	% 44	10008000
42	OCT1100000000000000,		10008100
43	"0ABS0000",	% 46	10008200
44	OCT0700010004100057,	% STD FN ID INLINE=#1, 1 PARAMETER	10008300
45	"0ATN0000",	% 48	10008400
46	OCT0700000070100061,	% STD FN INTRINSIC	10008500
47	"0CDN0000",	% 50	10008600
48	OCT0700010004000063,	% PSUEDO INTRINSIC NO. 1025	10008700
49	"0CDS0000",	% 52	10008800
50	OCT0700000064100065,	% STD FN INTRINSIC, 1 PARAMETER	10008900
51	"0DIV0000",	% 54	10009000
52	OCT3600000000003001,		10009100
53	"0EXP00C0",	% 56	10009200
54	OCT0700000100100071,	% STD FN INTRIN, 1 PARM,	10009300
55	"0IDN0000",	% 58	10009400
56	OCT0700010010000073,	% PSUEDO INTRINSIC NO. 1026	10009500
57	"0INT0000",	% 60	10009600

Data Documents/Inc.

OCT0700C1001410C075,	% STD FN, INLINE = #3, 1 PARM,	10009700
"OINV0000",	% 62	10009800
OCT0700C00610200077,	% STD FN INTRINSIC, 1 PARAMETER,	10009900
"OLOG0000",	% 64	10010000
OCT0700C000074100101,	% STD FN INTRINSIC, 1 PARAMETER,	10010100
"OMOD0000",	% 66	10010200
OCT3600C000000007001,		10010300
"ORND0000",	% 68	10010400
OCT0700C000621100105,	% STD FN INTRINSIC, 1 PARAMETER	10010500
"OSIN0000",	% 70	10010600
OCT0700C000060100107,	%	10010700
"OSQR0000",	% 72	10010800
OCT0700C000055100111,	% STD FN, INTRIN = #13, 1 PARM,	10010900
"OTAB0000",	% 74	10011000
OCT1600C000000100000,	% TABV	10011100
"OTAN0000",	% 76	10011200
OCT0700C000630100115,	% STD FN INTRINSIC, 1 PARAMETER	10011300
"OTRN0000",	% 78	10011400
OCT0700C000604200117,	% STD FN INTRINSIC, 2 PARAMETERS	10011500
"OZER0000",	% 80	10011600
OCT0700C010000000121,	% PSUEDU INTRINSIC NO. 1024	10011700
"OSGN0000",	% 82	10011800
OCT0700C010010100123,	% STD FN, INLINE = #2, 1 PARAMETER,	10011900
"OFILE000",	% 84	10012000
OCT2600C0000000000000,	% FILEV	10012100
"OGOT0000",	% 86	10012200
OCT1200C0000000000000,	% GQTDV	10012300
"OREAD000",	% 88	10012400
OCT2700C0000000000000,	% READV	10012500
"OSTEP000",	% 90	10012600
OCT1000C0000000000000,	% STEPV	10012700
"OTHENC00",	% 92	10012800
OCT2200C0000000000000,	% THENV	10012900
"OINPUT00",	% 94	10013000
OCT3000C0000000000000,	% INPUTV	10013100
"OPRINT00",	% 96	10013200
OCT3100C0000000000000,	%	10013300
"ORECORDS",	% 98	10013400
OCT1700C0000000000000;		10014500
FILL CTARRY[*] WITH		10015000
2, 16, 26, 36, 38, 40,	%NDX TO LAST WD OF SS GROUP	10015100
44, 82, 92, 96, 96, 98;	%NDX TO LAST WD OF IMBEDED WD GRP	10015200
SEQUENCER;		10015400
END.		20000000
END;END.	LAST CARD ON OCRDING TAPE	99999999

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

LABEL 00000000PRINTER00175099CC EX OBJECT/READ;FILE SUURCEFILE=SYMBOL/RESEQB;END+0000

OBJECT /READ

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57