

データベースの活用法(2) : DMPによるデータベース の検索について

河野, 重昭
九州大学教養部物理学教室

<https://doi.org/10.15017/1468128>

出版情報 : 九州大学大型計算機センター広報. 18 (6), pp. 487-496, 1985-11-25. 九州大学大型計算機センター
バージョン :
権利関係 :



データベースの活用法 (2)

DMPによるデータベースの検索について

河野 重昭*

1. はじめに

前回 [1] は、九州大学大型計算機センター研究開発部松尾文碩助教授らが1981年に開発したデータベース統合支援システムAdbis (A data base integration support) [2, 3, 4, 5] によるデータベースの構築について解説をした。

今回は、同Adbisの中のDMP (Data Manipulation Primitives) を用いたデータベースの検索について解説する。

テスト用データベースとしては、前回作成した "原子データベース" (TEST・ADBS) を使用する。

2. 検索例

検索用プログラムについては、後で述べることにして、先ず検索例を見てみよう。

```
EX REFER
*** REFER START ***
FORTRAN 77 COMPILER ENTERED
END OF COMPILATION
(ID/EL/A/Z/N/P/E)?Z
Z = ?1
3 DOCUMENT(S) FOUND
OUTPUT?(YES/NO)NO
(AND/OR/DIF/END)?AND
(ID/EL/A/Z/N/P/E)?N
N = ?1
1 DOCUMENT(S) FOUND
OUTPUT?(YES/NO)YES

  ID  EL   A   Z   N   P   E
  2   H   2   1   1   1   1
```

```
CONTINUE?(YES/NO)YES
(ID/EL/A/Z/N/P/E)?Z
Z = ?1
3 DOCUMENT(S) FOUND
OUTPUT?(YES/NO)NO
(AND/OR/DIF/END)?OR
(ID/EL/A/Z/N/P/E)?N
N = ?1
4 DOCUMENT(S) FOUND
OUTPUT?(YES/NO)YES
```

ID	EL	A	Z	N	P	E
1	H	1	1	0	1	1
2	H	2	1	1	1	1
3	H	3	1	2	1	1
4	HE	3	2	1	2	2

昭和60年9月14日受理

* 九州大学教養部物理学教室

```

CONTINUE?(YES/NO) YES
(ID/EL/A/Z/N/P/E)? Z
Z = ? 1
3 DOCUMENT(S) FOUND
OUTPUT?(YES/NO) NO
(AND/OR/DIF/END)? DIF
(ID/EL/A/Z/N/P/E)? N
N = ? 1
2 DOCUMENT(S) FOUND
OUTPUT?(YES/NO) YES

```

ID	EL	A	Z	N	P	E
1	H	1	1	0	1	1
3	H	3	1	2	1	1

```

CONTINUE?(YES/NO) NO
END OF GO, SEVERITY CODE=00
*** REFER NORMAL END ***

```

アンダーラインの部分はユーザーが入力する部分である。

EX REFER は、コマンドプロシジャ REFER.CLIST を起動させている。コマンドプロシジャ REFER.CLIST については次節で述べる。

(ID/EL/A/Z/N/P/E)? Z は、検索項目を聞いてきたので、Z(原子番号)を指定した。

Z = ? 1 は、原子番号を聞いてきたので、原子番号1を指定した。

3 DOCUMENT(S) FOUND は、3件見つかったことを表示している。

OUTPUT?(YES/NO) NO は、3件の内容を出力するかどうかを聞いてきたので、NO(出力しない)を指定した。

(AND/OR/DIF/END)? AND は、次に集合積を求める(AND)か、集合和を求める(OR)か、集合差を求める(DIF)か、何もしない(END)か聞いてきたので、AND(集合積を求める)を指定した。

(ID/EL/A/Z/N/P/E)? N は、更に検索項目を聞いてきたので、N(中性子の数)を指定した。

N = ? 1 は、中性子の数を聞いてきたので、1を指定した。

1 DOCUMENT(S) FOUND は、Z=1であり、そしてN=1である例が1件あることを表示している。

OUTPUT?(YES/NO) YES は、1件の内容を出力するかどうかを聞いてきたので、YES(出力する)を指定した。

次が出力例である。

ID	EL	A	Z	N	P	E
2	H	2	1	1	1	1

この後、集合和を求める(OR)例と集合差を求める(DIF)例が示されている。

3. コマンドプロシジャREFER.CLISTについて

先ず、コマンドプロシジャREFER.CLISTの内容を見てみよう。

```
LIST REFER.CLIST
KEQ52800I F9999.REFER.CLIST
00010 PROC 0
00020 CONTROL NOFLUSH PROMPT
00030 ERROR DO
00040 WRITE *** ERROR-E ***
00050 WRITE *** RC= &LASTCC ***
00060 EXIT
00070 END
00080 WRITE *** REFER START ***
00090 RUN REFER.FORT77 FIXED LIB ('QDB.LIBSL2')
00100 WRITE *** REFER NORMAL END ***
00110 EXIT
00120 END
KEQ52802I END OF DATA
```

10行のPROCは、コマンドプロシジャ文の開始を示し、0は位置パラメータもキーワードパラメータもないことを示している。

20行のCONTROLは、コマンドプロシジャの性格づけを行っている。NOFLUSHは、エラー発生時の処理を、PROMPTは、アテンション割込み発生の際の処理を可能にしている。

30-70行は、エラー処理の部分で、エラーが発生した時、エラーの表示と復帰コード&LASTCCの表示をし、EXITで計算を中止させている。

80-110行は、メインの部分で、"*** REFER START ***"の表示をし、検索用プログラムREFER.FORT77をRUNさせ、そして正常終了の場合に"*** REFER NORMAL END ***"を表示する。

90行のLIB('QDB.LIBSL2')は、DMPを使用する場合、必ず用いなければならない。

120行のENDは、コマンドプロシジャ文の終りを示す。

4. 検索プログラムREFER.FORT77について

検索プログラムREFER.FORT77のソース・リストを付録に示してあるが、ここでは要点だけを説明しておこう。

4.1 OPEN¥DとCLOSE¥D

Adbisを利用する場合、始めに OPEN¥D を、そして終りに CLOSE¥D を用いなければならない。ここでは、80行と1920行に用いてある。

4.2 ¥TPUTと¥TGET

110行の¥TPUTは TAC/LIB の中の1つで、端末へ出力するライブラリである。'(ID/EL/A/Z/N/P/E)?'の18文字を出力し、そこでストップし、次の入力待つ。

* 端末によっては、"\$"と表示されるので要注意。

120 行の ¥TGET も TAC/LIB の中の 1 つで、端末からの入力を読み取るライブラリである。JRC は入力文字の個数を示し、YN が文字を格納する。READ 文を用いると、"120 ?" が表示され、目ざわりであるが、¥TGET では、そのような表示をしない。

140 行の ¥ T P U T 文は，次と同じである．

```
PRINT *, ' *** INPUT ERROR '
```

4. 3 I C H A R

320行のICHARは、0～9までの文字を数字に変換するために用いてあるが、例えば“3”を変換すると243になるように、文字コードを変換すると240だけ大きい数になるので、240を引いてある。

340行は、10～99までの2桁の文字を2桁の数字に変換している。

4. 4 F O U T , F I N D ≠ D

370 行のFOUTは、検索件数を求めるサブルーチン(2100-2200 行)であり、検索件数KNOは、DMPのFINDYD(2140 行)により求めている。

IRC は、完了コードであり、0 でなければ異常終了ということである。

'TEST△△△△'(△は空白)の8文字は、検索する関数名である。

PRM1 を, ID = 3 として検索する場合について, 次を示す.

ID	EL	A	Z	N	P	E	ID	EL	A	Z	N	P	E
1	0	0	0	0	0	0	3	0	0	0	0	0	0

↑ ↓
 检索条件 检索值

IDについて検索するので、PRM1(1)=1とし、ID=3であるので、PRM1(8)=3とする(350-360行). 検索終了後は、もとに戻しておく(380-390行).

4 0 0 行は, 1 回目の時 $J J = 1$ で 1 1 0 行へ, 2 回目の時 $J J = 2$ で 1 2 0 行へとぶ.

4. 5 Q S B S T R

490 行の QSBSTR は TAC/LIB* の中の 1 つで、ここでは 4 バイト文字型 PEL の 4 文字を、4 バイト整数型 EL に代入している。EL=PEL とすると、S レベルのエラーが発生するので、QSBSTR を用いた。

4.6 SAVE≠D, MEET≠D, JOIN≠D, DIFF≠D, UNSA≠D

1530行のSAVEYDは、第1回目(JJ=1)のときにFINDYDにより検索したキー集合に'SET1△△△△'という名前をつけてセーブしている。

1570行のSAVEYDは、第2回目(JJ=2)のときにFINDYDにより検索したキー集合に'SET2△△△△'という名前をつけてセーブしている。

1600行のMEET YDは、セーブされた'SET1△△△△'と'SET2△△△△'のキー集合の集合積をとり、件数KNOを求めている。

* FACOM OS IV/F4 TAC/LIB解説書参照.

1 6 4 0 行のJOIN YDは、セーブされた 'SET 1 △△△△' と 'SET 2 △△△△' のキー集合の集合和をとり、件数KNOを求めている。

1 6 8 0 行のDIFF YDは、セーブされた 'SET 1 △△△△' と 'SET 2 △△△△' のキー集合の集合差をとり、件数KNOを求めている。

1 8 5 0 行のUNSA YDは、'SET 1 △△△△' のキー集合を消去している。

1 8 7 0 行のUNSA YDは、'SET 2 △△△△' のキー集合を消去している。

4. 7 DOUT, EVAL YD

1 3 7 0 行と1 7 7 0 行のDOUTは、検索したキー集合よりEVAL YD (2 2 5 0 行)を用いて値集合を求め、項目とその値を出力している。

2 2 5 0 行のEVAL YDのIRCは完了コードであり、'△△△△△△△△'は現キー集合の名前であり、IPRMは次のようになっている。項目の値が全部"1"になっているのは、全ての項目に対する値を求めるためである。

ID	EL	A	Z	N	P	E
1	1	1	1	1	1	1
1	2	3	4	5	6	7

↑ 項 目 の 値 ↑

BUFの内容は、検索件数が4件の場合、次のようになっている。

項目 ID EL A Z N P E																		
77	0	4	4	7	'I4'	4	'C4'	4	'I4'	4	'I4'	4	'I4'	4	'I4'	4	'I4'	4
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19

(くり返し)

エン트리件数が21あるので、 $19 + 7 \text{ (項目)} \times 21 \text{ (件)} = 166$ でよいが、余裕を見てBUF (200)とした(2230行)。

EVAL YDの領域は、4 バイト $\times$ 200 = 800 バイトとした。

4. 8 まとめ

8 0 行	—	9 0 行	:	オープン処理
1 0 0	—	2 5 0	:	検索項目の処理
2 6 0	—	4 0 0	:	IDコードによる検索
4 1 0	—	5 5 0	:	元素名(EL)による検索
5 6 0	—	7 0 0	:	原子量(A)による検索

7 1 0	—	8 5 0	:	原子番号 (Z) による検索
8 6 0	—	1 0 0 0	:	中性子の数 (N) による検索
1 0 1 0	—	1 1 5 0	:	陽子の数 (P) による検索
1 1 6 0	—	1 3 0 0	:	電子の数 (E) による検索
1 3 1 0	—	1 3 7 0	:	内容出力の処理
1 3 8 0	—	1 7 0 0	:	集合積, 集合和, 集合差の処理
1 7 1 0	—	1 7 7 0	:	内容出力の処理
1 7 8 0	—	1 9 1 0	:	コンティニューの処理
1 9 2 0	—	1 9 4 0	:	クローズ処理
1 9 5 0	—	2 0 9 0	:	エラー表示
2 1 0 0	—	2 2 0 0	:	検索件数の処理
2 2 1 0	—	2 3 9 0	:	内容出力の処理

5. おわりに

ここで用いた手法は, XDT用統計プログラム [6] で用いた手法と同じであり, これから SAS /GRAPH [7] につないで, グラフ化することもできる.

エントリー件数の少ないデータベースの検索としては, これで十分であるが, 次回から九州大学大型計算機センター研究開発部の方々にお願いして, "結晶構造データベースシステムXDT" [8] の検索に用いられた, より高度な検索方法についての解説をして頂く予定である.

参考文献

1. 河野 データベースの活用法 (1) —Adbis によるデータベースの構築について—, 九大大型計算機センター広報, **18**, 6, 1985, 482—486
2. 松尾, 二村, 高木 データベース統合支援システムAdbis (1) —Adbis の概要と DMP の仕様—, 同上, **14**, 2, 1981, 172—217.
3. 二村, 松尾, 高木 同 (2) —DMP のユーティリティプログラムと機能追加—同上, **14**, 3, 1981, 388—398.
4. 松尾, 高木, 二村 同 (3) —ホーン集合と HSI の仕様—, 同上, **14**, 4, 1981, 525—549.
5. 松尾, 二村, 高木 同 (4) —ホーン集合についての補足—, 同上, **15**, 1, 1982, 146—153.
6. 河野 XDT用統計プログラムについて, 同上, **17**, 2, 1984, 67—74.
7. 武富, 川崎, 遠矢, 柳池, 景川 SAS/GRAPHの使用について, 同上, **15**, 4, 1982, 367—397.
8. 河野, 高木, 松尾, 二村, 鬼塚 結晶構造データベースシステムXDTについて (総合版), 同上, **16**, 6, 1983, 556—604.

付録. 検索用プログラム

```

LIST REFER.FORT77
KEQ52800I F7777.REFER.FORT77
00010 C    PROGRAM OF REFERENCE FOR TEST DATABASE
00020 C    CODED BY S.KAWANO
00030      PROGRAM REFER
00040      IMPLICIT INTEGER*4(A-Z)
00050      INTEGER PRM1(14)/14*0/
00060      CHARACTER PEL*4,YN(136)/136*1H /
00070      COMMON KNO
00080      CALL OPEN#D(IRC,'ADBS      ','F7777      ','MSTR      ')
00090      IF(IRC.NE.0) GOTO 901
00100      JJ=1
00110 10 CALL $TPUT(18,'(ID/EL/A/Z/N/P/E)?',1)
00120      CALL $TGET(JRC,YN)
00130      IF(JRC.NE.0) GOTO 13
00140      CALL $TPUT(15,'*** INPUT ERROR',0)
00150      GOTO 10
00160 13 IF(YN(1).EQ.'I') GOTO 20
00170      IF(YN(1).EQ.'E') GOTO 15
00180      IF(YN(1).EQ.'A') GOTO 40
00190      IF(YN(1).EQ.'Z') GOTO 50
00200      IF(YN(1).EQ.'N') GOTO 60
00210      IF(YN(1).EQ.'P') GOTO 70
00220      GOTO 10
00230 15 IF(YN(2).EQ.'L') GOTO 30
00240      IF(YN(2).EQ.' ') GOTO 80
00250      GOTO 10
00260 20 CALL $TPUT(6,'ID = ?',1)
00270      CALL $TGET(JRC,YN)
00280      IF(JRC.NE.0) GOTO 23
00290      CALL $TPUT(15,'*** INPUT ERROR',0)
00300      GOTO 20
00310 23 IF(YN(2).NE.' ') GOTO 25
00320      II=ICHAR(YN(1))-240
00330      GOTO 28
00340 25 II=10*(ICHAR(YN(1))-240)+ICHAR(YN(2))-240
00350 28 PRM1(1)=1
00360      PRM1(8)=II
00370      CALL FOUT(PRM1,JJ)
00380      PRM1(1)=0
00390      PRM1(8)=0
00400      GOTO (110,120),JJ
00410 30 CALL $TPUT(6,'EL = ?',1)
00420      CALL $TGET(JRC,YN)
00430      IF(JRC.NE.0) GOTO 33
00440      CALL $TPUT(15,'*** INPUT ERROR',0)
00450      GOTO 30
00460 33 PEL(1:2)=' '
00470      PEL(3:3)=YN(1)
00480      PEL(4:4)=YN(2)
00490      CALL QSBSTR(EL,PEL,4)
00500      PRM1(2)=1
00510      PRM1(9)=EL
00520      CALL FOUT(PRM1,JJ)
00530      PRM1(2)=0
00540      PRM1(9)=0
00550      GOTO (110,120),JJ
00560 40 CALL $TPUT(5,'A = ?',1)
00570      CALL $TGET(JRC,YN)
00580      IF(JRC.NE.0) GOTO 43
00590      CALL $TPUT(15,'*** INPUT ERROR',0)

```

```

00600      GOTO 40
00610 43 IF (YN(2).NE.' ') GOTO 45
00620      II=ICHAR(YN(1))-240
00630      GOTO 48
00640 45 II=10*(ICHAR(YN(1))-240)+ICHAR(YN(2))-240
00650 48 PRM1(3)=1
00660      PRM1(10)=II
00670      CALL FOUT(PRM1,JJ)
00680      PRM1(3)=0
00690      PRM1(10)=0
00700      GOTO (110,120),JJ
00710 50 CALL $TPUT(5,'Z = ?',1)
00720      CALL $TGET(JRC,YN)
00730      IF (JRC.NE.0) GOTO 53
00740      CALL $TPUT(15,'*** INPUT ERROR',0)
00750      GOTO 50
00760 53 IF (YN(2).NE.' ') GOTO 55
00770      II=ICHAR(YN(1))-240
00780      GOTO 58
00790 55 II=10*(ICHAR(YN(1))-240)+ICHAR(YN(2))-240
00800 58 PRM1(4)=1
00810      PRM1(11)=II
00820      CALL FOUT(PRM1,JJ)
00830      PRM1(4)=0
00840      PRM1(11)=0
00850      GOTO (110,120),JJ
00860 60 CALL $TPUT(5,'N = ?',1)
00870      CALL $TGET(JRC,YN)
00880      IF (JRC.NE.0) GOTO 63
00890      CALL $TPUT(15,'*** INPUT ERROR',0)
00900      GOTO 60
00910 63 IF (YN(2).NE.' ') GOTO 65
00920      II=ICHAR(YN(1))-240
00930      GOTO 68
00940 65 II=10*(ICHAR(YN(1))-240)+ICHAR(YN(2))-240
00950 68 PRM1(5)=1
00960      PRM1(12)=II
00970      CALL FOUT(PRM1,JJ)
00980      PRM1(5)=0
00990      PRM1(12)=0
01000      GOTO (110,120),JJ
01010 70 CALL $TPUT(5,'P = ?',1)
01020      CALL $TGET(JRC,YN)
01030      IF (JRC.NE.0) GOTO 73
01040      CALL $TPUT(15,'*** INPUT ERROR',0)
01050      GOTO 70
01060 73 IF (YN(2).NE.' ') GOTO 75
01070      II=ICHAR(YN(1))-240
01080      GOTO 78
01090 75 II=10*(ICHAR(YN(1))-240)+ICHAR(YN(2))-240
01100 78 PRM1(6)=1
01110      PRM1(13)=II
01120      CALL FOUT(PRM1,JJ)
01130      PRM1(6)=0
01140      PRM1(13)=0
01150      GOTO (110,120),JJ
01160 80 CALL $TPUT(5,'E = ?',1)
01170      CALL $TGET(JRC,YN)
01180      IF (JRC.NE.0) GOTO 83
01190      CALL $TPUT(15,'*** INPUT ERROR',0)
01200      GOTO 80
01210 83 IF (YN(2).NE.' ') GOTO 85
01220      II=ICHAR(YN(1))-240

```

```

01230      GOTO 88
01240      85 II=10*(ICHAR(YN(1))-240)+ICHAR(YN(2))-240
01250      88 PRM1(7)=1
01260          PRM1(13)=II
01270          CALL FOUT(PRM1,JJ)
01280          PRM1(7)=0
01290          PRM1(14)=0
01300          GOTO (110,120),JJ
01310      110 CALL $TPUT(15,'OUTPUT?(YES/NO)',1)
01320          CALL $TGET(JRC,YN)
01330          IF(JRC.NE.0) GOTO 111
01340          CALL $TPUT(15,'*** INPUT ERROR',0)
01350          GOTO 110
01360      111 IF(YN(1).EQ.'N') GOTO 112
01370          CALL DOUT
01380      112 CALL $TPUT(17,'(AND/OR/DIF/END)?',1)
01390          CALL $TGET(JRC,YN)
01400          IF(JRC.NE.0) GOTO 113
01410          CALL $TPUT(15,'*** INPUT ERROR',0)
01420          GOTO 110
01430      113 IF(YN(1).EQ.'E') GOTO 162
01440          IF(YN(1).EQ.'A') GOTO 114
01450          IF(YN(1).EQ.'O') GOTO 115
01460          IF(YN(1).EQ.'D') GOTO 116
01470          GOTO 110
01480      114 IJ=1
01490          GOTO 118
01500      115 IJ=2
01510          GOTO 118
01520      116 IJ=3
01530      118 CALL SAVE$D(IRC,'SET1      ')
01540          IF(IRC.NE.0) GOTO 902
01550          JJ=2
01560          GOTO 10
01570      120 CALL SAVE$D(IRC,'SET2      ')
01580          IF(IRC.NE.0) GOTO 902
01590          GOTO (130,140,150),IJ
01600      130 CALL MEET$D(IRC,KNO,'SET1      ','SET2      ')
01610          IF(IRC.NE.0) GOTO 904
01620          PRINT*,KNO,' DOCUMENT(S) FOUND'
01630          GOTO 160
01640      140 CALL JOIN$D(IRC,KNO,'SET1      ','SET2      ')
01650          IF(IRC.NE.0) GOTO 905
01660          PRINT*,KNO,' DOCUMENT(S) FOUND'
01670          GOTO 160
01680      150 CALL DIFF$D(IRC,KNO,'SET1      ','SET2      ')
01690          IF(IRC.NE.0) GOTO 906
01700          PRINT*,KNO,' DOCUMENT(S) FOUND'
01710      160 CALL $TPUT(15,'OUTPUT?(YES/NO)',1)
01720          CALL $TGET(JRC,YN)
01730          IF(JRC.NE.0) GOTO 161
01740          CALL $TPUT(15,'*** INPUT ERROR',0)
01750          GOTO 160
01760      161 IF(YN(1).EQ.'N') GOTO 162
01770          CALL DOUT
01780      162 CALL $TPUT(17,'CONTINUE?(YES/NO)',1)
01790          CALL $TGET(JRC,YN)
01800          IF(JRC.NE.0) GOTO 163
01810          CALL $TPUT(15,'*** INPUT ERROR',0)
01820          GOTO 160
01830      163 IF(YN(1).EQ.'N') GOTO 170
01840          IF(JJ.EQ.2) THEN
01850              CALL UNSA$D(IRC,'SET1      ')

```

```

01860      IF(IRC.NE.0) GOTO 907
01870      CALL UNSA%D(IRC,'SET2      ')
01880      IF(IRC.NE.0) GOTO 907
01890      JJ=1
01900      ENDIF
01910      GOTO 10
01920 170 CALL CLOS%D(IRC,'ADBS      ')
01930      IF(IRC.NE.0) GOTO 903
01940      STOP
01950 901 PRINT*,'*** OPEN%D  ERROR, IRC=',IRC
01960      STOP
01970 902 PRINT*,'*** SAVE%D  ERROR, IRC=',IRC
01980      STOP
01990 903 PRINT*,'*** CLOS%D  ERROR, IRC=',IRC
02000      STOP
02010 904 PRINT*,'*** MEET%D  ERROR, IRC=',IRC
02020      STOP
02030 905 PRINT*,'*** JOIN%D  ERROR, IRC=',IRC
02040      STOP
02050 906 PRINT*,'*** DIFF%D  ERROR, IRC=',IRC
02060      STOP
02070 907 PRINT*,'*** UNSA%D  ERROR, IRC=',IRC
02080      STOP
02090      END
02100 C    FIND%D AND PRINT OUT
02110      SUBROUTINE FOUT(PRM1,JJ)
02120      INTEGER*4 IRC,KNO,PRM1(14),JJ
02130      COMMON KNO
02140      CALL FIND%D(IRC,KNO,'TEST      ',PRM1)
02150      IF(IRC.NE.0) GOTO 900
02160      IF(JJ.EQ.1) PRINT*,KNO,' DOCUMENT(S) FOUND'
02170      RETURN
02180 900 PRINT*,'*** FIND%D  ERROR, IRC=',IRC
02190      STOP
02200      END
02210 C    PRINTOUT OF DATA
02220      SUBROUTINE DOUT
02230      INTEGER*4 IRC,BUF(200),IPRM(7)/7*1/,KNO
02240      COMMON KNO
02250      CALL EVAL%D(IRC,BUF,800,'      ',IPRM)
02260      IF(IRC.NE.0) GOTO 900
02270      PRINT*,' '
02280      PRINT*,' ID EL  A  Z  N  P  E'
02290      I1=13
02300      DO 10 I=1,KNO
02310      I1=I1+7
02320      WRITE(*,100) (BUF(J),J=I1,I1+6)
02330 100 FORMAT(1H ,I4,A4,5I4)
02340 10 CONTINUE
02350      PRINT*,' '
02360      RETURN
02370 900 PRINT*,'*** EVAL%D  ERROR, IRC=',IRC
02380      STOP
02390      END
KEQ52802I  END OF DATA

```